

INPUT
Publicación práctica para usuarios de
sinclair
Revista mensual 1986
Precio 350 Ptas
Año 1 Número 11
DETECCION DE OBJETOS EN PANTALLA
GRAFICOS CON JOYSTICK
EL SONIDO DEL SPECTRUM
ZX Spectrum+

CAULDRON · II



El Hechizo

DISTRIBUIDOR EXCLUSIVO PARA ESPAÑA: ERBE SOFTWARE C/. STA. ENGRACIA, 17
28010 MADRID. TEL. (91) 447 34 10 • DELEGACION BARCELONA, AVDA. MISTRAL, N.° 10 • TEL. (93) 432 07 31



AÑO 1 NUMERO 11

DIRECTOR:
Alejandro Diges
COORDINADOR EDITORIAL:
Francisco de Molina
DISEÑO GRAFICO:
Tomás López
COLABORADORES:
Antonio Taratiel, Luis R. Palencia, Francisco Tórtola, Benito Román, Esther de la Cal, Ernesto del Valle, Equipo Molisoft.

INPUT Sinclair es una publicación juvenil de EDICIONES FORUM

GERENTE DIVISION DE REVISTAS:
Angel Sabat

PUBLICIDAD: José Real-Grupo Jota
Madrid: c/ General Varela, 35
Teléf. 270 47 02/03
Barcelona: Avda. de Sarriá, 11-13, 1.º
Teléf. 250 23 99

FOTOMECANICA: Ochoa, S. A.

COMPOSICION: EFCA, S. A.

IMPRESION: Sirven Grafic
C/ Gran Vía, 754-756. 08013 Barcelona
Depósito legal: B-21954-1986

SUSCRIPCIONES: EDISA,
López de Hoyos, 141. 28002 Madrid
Teléf. (91) 415 97 12

REDACCION:
Alberto Alcocer, 46, 4.º
28016 Madrid. Teléf. 250 10 00

DISTRIBUIDORA
R.B.A. PROMOTORA DE EDICIONES, S. A.
Travesera de Gracia, 56. Edificio Odiseus
08006 Barcelona.

El precio será el mismo para Canarias que para la Peninsula y en él irá incluida la sobretasa aérea.

Se ha solicitado el control OJD

INPUT Sinclair es independiente y no está vinculada a Sinclair Research o sus distribuidores.

INPUT no mantiene correspondencia con sus lectores, si bien la recibe, no responsabilizándose de su pérdida o extravío. Las respuestas se canalizarán a traves de las secciones adecuadas en estas páginas.

Copyright ilustraciones del fondo gráfico de Marshall Cavendish, págs. 8, 9, 10, 11, 23, 26, 27, 28, 31, 32, 33, 34, 36, 37, 39, 40, 41, 51, 53.

INPUT sinclair

SUMARIO

EDITORIAL — 4

ACTUALIDAD — 6

CODIGO MAQUINA
EL SONIDO DEL SPECTRUM — **8**
RUTINAS EN C.M. — **44**

APLICACIONES
PREDICIENDO LO IMPREDECIBLE — **23**

ENTREVISTA
LOS DINAMIC CAMBIAN DE MANSION — **20**

PROGRAMACION
GRAFICOS CON JOYSTICK — **16**
DETECCION DE OBJETOS EN LA PANTALLA — **39**
COMO SE ALMACENAN LOS PROGRAMAS — **50**

REVISTA DE SOFTWARE — 54

PROGRAMACION DE JUEGOS (COLECCIONABLE) — 31
HACER SALTAR LA BANCA
CONVIERTE TU AVENTURA EN UNA EPOPEYA

¿PARA CUANDO EL NUEVO 128?

Ha pasado suficiente tiempo desde que **Alan Sugar** se hiciera con la exclusividad para la distribución de los productos desarrollados por **Sinclair.** Pero las noticias en torno al futuro continúan siendo confusas. Parece que en nuestro país habrá reestructuraciones importantes en la distribución.

En las islas británicas se dice que el **Pandora** sigue en la línea de espera y que no es un producto demasiado atractivo para el usuario en su actual concepción. También se baraja septiembre como el momento idóneo para lanzar el rumoreado **Sinclair 128** con unidad de *cassette* incorporada en la carcasa. Se trataría de una máquina de juegos más sofistificada y económica en precio.

Por otro lado, **Amstrad** ya ha mostrado su compatible **PC** y lo que más nos asombra es su precio extremadamente bajo. Con una unidad de *diskettes* y 256 Kbytes de RAM, costaría menos de 500 libras esterlinas (¡unas 110.000 ptas!).

Estos movimientos pueden representar una profunda dicotomización del mercado de usuarios con bajo poder adquisitivo. Por un lado los que desean un sistema profesional y los preocupados por los juegos en máquina de altas prestaciones.

LOS MEJORES DE INPUT

Hemos pensado que es interesante disponer de un ***ranking*** que ponga en claro, mes a mes, cuáles son los programas preferidos de nuestros lectores. Para ello, es obligado preguntaros directamente y tener así el mejor termómetro para conocer vuestras preferencias. Podéis votar por cualquier programa aunque no haya sido comentado todavía en **INPUT**.

El resultado de las votaciones será publicado en cada número de **INPUT**.

Entre los votantes sortearemos 10 cintas de los títulos que pidáis en vuestros cupones.

Nota: No es preciso que cortéis la revista, una copia hecha a máquina o una simple fotocopia sirven.

Enviad vuestros votos a: **LOS MEJORES DE INPUT** Alberto Alcocer, 46 - 4.° B. 28016 Madrid

ELIGE TUS PROGRAMAS

Primer título elegido

Segundo título elegido

Tercer título elegido

Programa que te gustaría conseguir

Qué ordenador tienes

Nombre

1.er Apellido

2.° Apellido

Fecha de nacimiento

Teléfono

Dirección

Localidad

Provincia

EQUINOX

Spectrum/Amstrad £9.95

MIKRO-GEN

ERBE Software

OLIVER FREY

DISTRIBUIDOR EXCLUSIVO ERBE SOFTWARE C/. STA. ENGRACIA, 17 · 28010 MADRID, TFNO. (91) 447 34 10
DELEGACION BARCELONA, AVDA. MISTRAL, N° 10, TFNO. (93) 432 07 31



DE PELICULA

Una reciente nota de prensa confirma que Activision ha conseguido los derechos exclusivos para desarrollar programas relacionados con importantes producciones cinematográficas por venír. Entre los titulos se hallan Laberinto, coproducida por Lucas Films, Howard el Pato, Aliens y Probleмones en la Pequeña China.
Por otro lado, esta firma acaba de lanzar el programa Hacker II: The Doomsday Papers, del que se dice que ofrece aún mas sorpresas que su antecesor, Hacker.



ORDENADOR FABRICADO EN ESPAÑA

No es nuestra costumbre mencionar en esta sección equipos de otras marcas. A no ser que vengan precedidos por algún mérito especial.
Este es el caso del compatible PC que fabrica Promax y comercializa Multihard. Ambas son dos firmas nacionales, que presentaron su estrategia durante el pasado Informat.
No es la primera vez que una firma española desarrolla un ordenador de esta clase. Otras iniciativas han tenido distintos grados de fortuna.
Promax es una empresa fabricante de instrumentos electrónicos de medida, que goza de excelente reputación mas allá de nuestras fronteras.
Confiamos en que este sea el principio de un fructifero camino que lleve a nuevas lineas de fabricación.



MAS CARRERAS DE MOTOS

Un increible programa de carreras de motos para Spectrum está a punto de aparecer. Lleva por título T.T. Racer y para que sus efectos fuesen lo mas reales posible, el equipo de desarrollo colocó una camara de video en el depósito de una moto de gran cilindrada. después intentaron pasar la pelicula a dibujos de ordenador.
A juzgar por lo visto, este puede ser uno de los programas con mayor nivel entre los de su clase.



HACKERS IN FRAGANTI

Dos *Hackers*, nombre con elque se bautiza a los intrusos informáticos, acaban de ser condenados en el Reino Unido tras la denuncia presentada por British Telecom. Ambos accedian a un ordenador privado, despues de haber dado con la clave de acceso al mismo.
No se trata de inexpertos, pues uno de ellos escribe regularmente en una conocida publicación británica de ordenadores y el otro proporciona servicios a través de la red Micronet.
Es la primera vez que allí se produce condena por este tipo de hechos. La curiosa sentencia hace referencia a "utilizar un dispositivo que registra o guarda información por medios electrónicos con intención de inducir a un ordenador a que le acepte como genuíno, derivandose de ello un acto que perjudique a los dueños u operadores de ese ordenador". Todo un precedente.

LOS DIEZ GANADORES DEL CONCURSO INPUT/ANAYA MULTIMEDIA

A continuación mostramos la lista de los diez elegidos entre quienes habeís respodido a la primera convocatoria del concurso.

* Gregorio Morell Rodriguez. Alcorcon. Madrid.
* Pedro Pablo Perez Cortes. Malaga.
* Javier Martinez Cormenzana. Bilbao.
* Gabriel Ortas Gonzalez. Madrid.
* Manuel Albert Ostal. Torreblanca. Castellon.
* Pindaro Martin Orihuela. Madrid.
* Jose Ignacio Gude Basterrechea. Madrid.
* Albert Colome Mallolas. Barcelona.
* Vicente Perez Gracia. Sta. Cruz de Tenerife. Tenerife.
* Victor Manuel Gonzalez Munoz. Madrid.

En breve plazo recibireis una carta notificandoos el premio, junto con el catálogo de titulos de la editorial para que elijais los deseados. Gracias por vuestra participación.

Ordena tus propias ideas

Le sacarás partido a tu ordenador



TU PRIMER LIBRO DEL ZX SPECTRUM.
J. Dewhirst y R. Tennison
848 ptas.



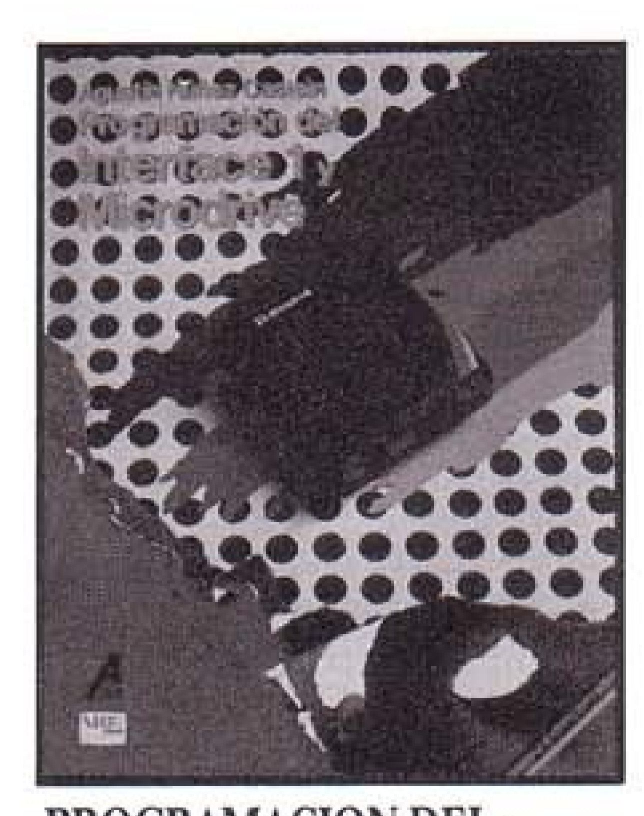
"SPRITES" Y GRAFICOS EN LENGUAJE MAQUINA. (ZX SPECTRUM)
John Durst
1.537 ptas.



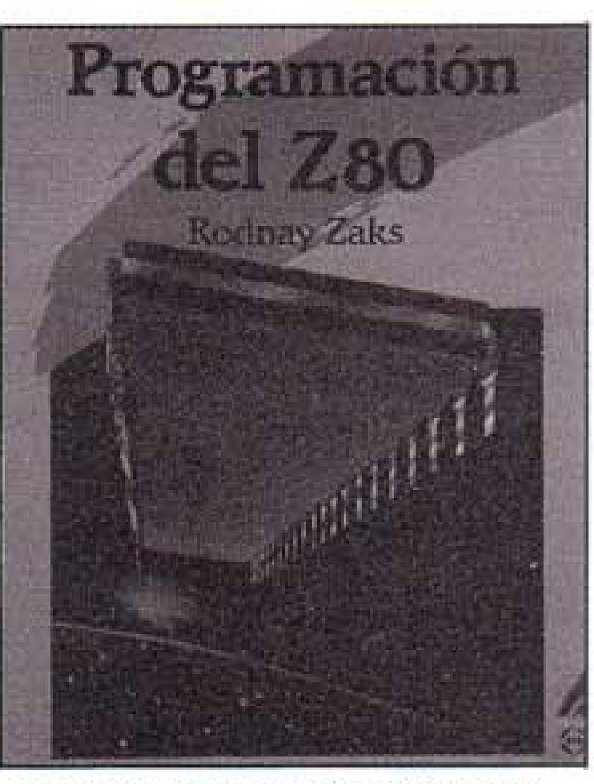
EL LIBRO GIGANTE DE LOS JUEGOS PARA ZX SPECTRUM
Tim Hartnell
1.431 ptas.



JUEGOS GRAFICOS DE AVENTURA. Técnicas de diseño.
Richard Hurley
1.484 ptas.



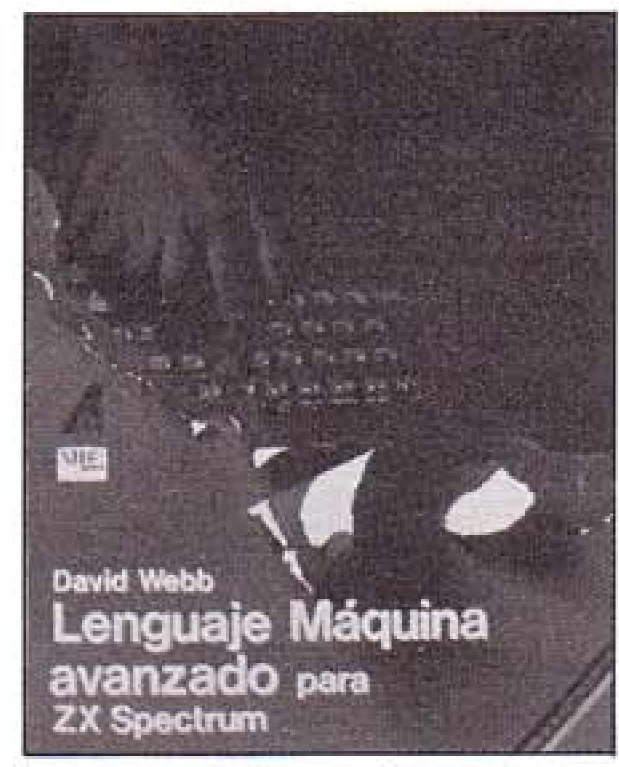
PROGRAMACION DEL INTERFACE I Y MICRODRIVE.
Agustín Núñez
1.166 ptas.



PROGRAMACION DEL Z80
Rodnay Zaks
2.915 ptas.



DISEÑO DE GRAFICOS Y VIDEOJUEGOS. TRATAMIENTO EN TRES DIMENSIONES.
(Incluye cassette)
Ian O. Angel y Brian J. Jones
3.392 ptas.



LENGUAJE MAQUINA AVANZADO PARA ZX SPECTRUM
David Webb
1.484 ptas.

☐ Les ruego me envíen el catálogo de su editorial.

☐ Les ruego me envíen los siguientes títulos:

____

____

TOTAL __________

☐ Adjunto talón bancario a
GRUPO DISTRIBUIDOR EDITORIAL, S. A.

☐ Pagaré contrarrembolso (+ 125 pesetas de gasto de envío).

☐ Giro postal.

Nombre ____________________

Profesión ____________________

Dirección ____________________

C. P. __________ Localidad __________

Provincia ____________________

IS



Adquiéralos en su librería habitual.
Si no le es posible o desea que le enviemos nuestro catálogo, envíe este cupón a:
Apdo. de Correos 14632, Ref. D. de C. 28080 MADRID

ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA ANAYA

EL SONIDO DEL SPECTRUM

- ¿QUE ES UN PORT?
- FIJACION DE LA ALTURA DE LOS SONIDOS
- TIEMPO CORRECTO
- SINTONIA FINA Y GRUESA

Con el comando BEEP del BASIC, los efectos sonoros obtenibles con el Spectrum **son bastante limitados. Pero con el comando OUT en código máquina puedes imitar sirenas y disparos con rayos laser, además de conseguir un borde de pantalla de siete colores.**

En condiciones normales el microprocesador direcciona la memoria del ordenador. Con muchos micros domésticos, si quieres acceder a un dispositivo exterior, tal como una impresora, un televisor o incluso el teclado del propio ordenador, tienes que hacerlo a través de una dirección de memoria asociada con un *port* de salida. Sin embargo con ordenadores soportados por un **Z80**, tal como el **Spectrum**, puedes acceder directamente a los *ports*. Puedes hacer esto de dos maneras: bien en BASIC con los comandos IN y OUT o en código máquina con los mnemónicos de lenguaje ensamblador **in** y **out**.

¿QUE ES UN PORT?

Un *port* es un canal de comunicaciones entre el ordenador y el mundo exterior, y al referirse al mundo exterior se incluye también el teclado, que es un periférico del sistema formado por el microprocesador con sus memorias RAM y ROM asociadas.

Ya conoces la manera de utilizar el comando IN del BASIC en relación con el uso de un *joystick*, y cómo se puede utilizar el lenguaje ensamblador para acceder al teclado.

Los comandos OUT y out trabajan de una forma muy parecida; envían datos a los periféricos pero no los reciben de ellos. Se pueden usar para controlar el borde de la pantalla de televisión, y para emitir sonidos por el altavoz del **Spectrum** de una forma mucho más manejable que con el comando BEEP del BASIC. Aquí nos ocuparemos de examinar el comando out del código máquina, ya que el OUT del BASIC es bastante lento.

Debido a la forma en que se utiliza out para generar el sonido, se hace necesaria la velocidad del código máquina. Lo que haces es conectar y desconectar el altavoz. Si haces esto una vez se produce un *clic*, parecido al chas-



quido que a veces se oye al encender un tocadiscos. El truco consiste en hacer esto repetidamente y con mucha rapidez. Si los *clics* del altavoz se van haciendo cada vez más rápidos, la sucesión de chasquidos se convertirá en una zumbido grave; si todavía se hace más rápido, se elevará el tono del sonido resultante.

SONIDO CON out

Con la siguiente rutina en lenguaje máquina se obtiene un sonido cuyo tono se va elevando. Teclea CLEAR 64599, y a continuación lo siguiente:

```
         org 64600
         ld a,(23624)
         rrca
         rrca
         rrca
         ld b,0
bucle    push bc
         xor 16
         out 254,a
pausa    nop
         nop
         djnz pausa
         pop bc
         djnz bucle
              ret
```

El *port* 254 es el que controla el altavoz del **Spectrum**. Pero este mismo *port* es el que controla también el borde de pantalla del televisor. Al emitir un sonido con ayuda de **out**, no querrás que cambie tambien el color del borde de la pantalla, por lo que tienes que hacer una pequeña rutina que deje inalterado dicho borde.

En primer lugar, carga el contador con el contenido de la variable del sistema que hay en la dirección de memoria 23624. Los colores del **Spectrum** se especifican con un número del 0 al 7. Normalmente los bits que controlan el color del borde de pantalla son los bits 3,4 y 5. Pero cuando se controla a través del *port* 254, son los bits 0, 1 y 2 los que temporalmente modifican el color del borde de pantalla.

Así pues, para asegurarte de que el color del borde de patalla no cambia durante el efecto sonoro, tienes que desplazar tres lugares hacia la derecha los bits 3, 4 y 5 (a las posiciones 0, 1 y 2).

DESPLAZAMIENTOS Y ROTACIONES

Aunque hay varios comandos que pueden hacer esto, aquí hemos elegido el comando **rrca** (rotación en el acumulador con acarreo hacia la derecha). Con esta instrucción, todos los bits del acumulador se mueven un lugar hacia la derecha. Se llama «rotación» debido a que el contenido del bit 0 va a situarse al bit 7. En cambio el desplazamiento simple mueve todos los bits un lugar hacia la izquierda o hacia la derecha,llenando con un cero el bit que queda disponible. Los desplazamientos se usan en las operaciones aritméticas. Como puedes ver fácilmente, un desplazamiento a la izquierda equivale a multiplicar un número por dos, mientras que un desplazamiento a la derecha equivale a la división por dos.

Sin embargo en este caso hemos utilizado una rotación, debido a que sólo hay que desplazar tres de los bits. El contenido de los otros bits no importa. La instrucción **rrca** también copia el contenido del bit cero en el indicador de acarreo (*carry flag*). Pero en este caso no tiene importancia el indicador de acarreo, ya que no se va a examinar para nada.

Para desplazar tres lugares a la derecha los bits de color del borde de pantalla, se ejecuta tres veces la instrucción **rrca**. El resultado es que el valor correspondiente al color del borde de pantalla queda dividido por ocho.

Pero ahora que el color del borde de pantalla está almacenado en los bits 3, 4 y 5, el valor del color (que es un número comprendido entre 0 y 7) ya está multiplicado por ocho.

Por eso cuando se utiliza ahora el comando **out**, quedará especificado el mismo color de borde que antes y no se detectará ningún cambio.

POSICION DE LOS CONTADORES

Vamos a utilizar el registro B como un doble contador. Para empezar, se carga con el valor 0, y este valor se copia en el *stack* (pila) mediante la instrucción **push bc**. El contenido del registro B no puede transferirse al *stack* por separado. Las instrucciones **push** y **pop** de transferencia al *stack* sólo trabajan sobre pares de registros, por lo que el contenido de B se transfiere al *stack* junto con el contenido del registro C. Pero como no vas a hacer nada con C, no afectará al programa.

GENERACION DE SONIDOS

El bit 4 del *port* 254 controla el altavoz del **Spectrum**. Si se cambia este

bit, el diafragma del altavoz se mueve hacia adentro y hacia afuera generando un sonido. El bit se cambia haciendo una operación lógica de «OR exclusiva» con el valor 16. Así si el valor de dicho bit está puesto a 1, pasa a valer 0; si su valor es 0, pasa a valer 1.

A continuación la instrucción **out 254,a** saca el contenido del acumulador a través del *port* 254. (Fíjate bien que en la mayoría de los ensambladores comerciales las instrucciones **in** y **out** requieren que el número del *port* vaya entre paréntesis, por lo que si no utilizas un programa ensamblador, tendrás que escribir dicha instrucción como **out(254),a**).

DANDO EL TONO

La instrucción **nop** significa no operación. No hace nada en absoluto y su código de operación en hexadecimal es 00. Sin embargo se requiere algún tiempo para ejecutar dicha instrucción, aproximadamente 1 microsegundo, es decir la millonésima parte de un segundo. La razón de utilizar aquí dicha instrucción es que hace que el procesador ejecute este bucle más lentamente. La velocidad con la que el microprocesador recorre el bucle controla la velocidad con que se moverá el diafragma del altavoz, y en consecuencia la altura del sonido resultante.

Pero como puedes ver los **nop**'s no se ejecutan sólo dos veces. La instrucción **djnz** (decremento y salto si el resultado no es cero), hace que el bucle pausa se ejecute muchas veces.

La instrucción **djnz** opera sobre el registro B. Por tanto, la primera vez que se recorre el bucle se decrementa el valor de B desde 0 a 255. Seguidamente recorre el bucle otras 255 veces hasta que se alcanza nuevamente el valor 0.

Una vez que el programa sale del bucle de pausa el último valor se transfiere con la instrucción **pop** desde el *stack* al par de registros BC. Con esto se restaura el valor del registro B, seguidamente la instrucción **djnz** decrementa de nuevo y envía al programa de nuevo al bucle. El contenido del par de registros BC vuelve a pasar de nuevo al *stack* con la instrucción **push**. De esta forma, cada vez que el procesador recorre el citado bucle, se decrementa el contador del *stack* y el valor de partida del bucle pausa , que está en el contador B, se decrementa cada vez en uno. El resultado es que pausa se ejecuta una vez menos y el tono del sonido sube.

BORDE DE PANTALLA DE SIETE COLORES

Cuando especificas el color de BORDER desde el BASIC, le asignas un número entre 0 y 7. El color resultante del borde de pantalla se corresponde entonces con el número asignado.

La siguiente rutina utiliza en cambio el comando **out** de código máquina para obtener un borde de pantalla de siete colores. Podría conseguirse un borde de pantalla de ocho colores, pero no tiene interés el que uno de los colores de borde de pantalla sea el mismo que el de la pantalla.

```
        org 64600
repe    halt
        xor a
bucle   out 254,a
        ld b,205
pausa   ld e,2
inter   dec e
        jr nz,inter
        djnz pausa
        ld d,a
        ld a,$7F
        in a,254
        rra
        rts nc
        ld a,d
        inc a
        cp 7
        jr nz,bucle
        jr repe
```

SINCRONIZACION DE LA PANTALLA

La instrucción **halt** hace que el microprocesador se quede esperando





hasta que se produzca una interrupción; cuando ésta llega, envía el control a la siguiente instrucción. Con el **Spectrum**, la interrupción se produce por la exploración de la pantalla. Así, **halt** arranca esta subrutina cuando empieza el barrido en la parte superior de la pantalla, sincronizando las bandas de los bordes de pantalla con el borde de la imagen de televisión.

La instrucción **xor a** realiza una operación lógica de OR exclusiva consigo mismo, lo cual es una manera rápida de ponerlo a 0, y a este 0 se le da salida con la instrucción **out** a través del *port* 254. En código máquina cero significa, igual que en BASIC, negro. Por eso la parte superior de la pantalla empieza con negro.

AJUSTE DE LA PAUSA

En esta rutina el control estricto de los tiempos es crítico. La longitud de la pausa sirve para especificar la anchura de las bandas de color del borde de pantalla. Por eso la pausa está aquí controlada por dos bucles, el bucle interior (inter) que establece el retardo de una forma basta y el bucle exterior (pausa) que constituye un ajuste fino del retardo.

El bucle interior se ejecuta sobre el registro E, que se carga con el valor 2 con la instrucción **ld e,2** y se decrementa con la instrucción **dec e**. A continuación la instrucción **jrnz,inter** realiza un salto relativo hacia atrás hasta **inter** si el resultado no es cero. De esta manera el bucle interior se ejecuta dos veces cada vez que el procesador recorre el bucle exterior.

El bucle exterior se ejecuta 205 veces con el registro B. Está formado por la instrucción **ld b,205** que carga dicho número en el registro B y la **djnz**, que decrementa el registro B y salta si el resultado no es cero. No hay una instrucción similar que trabaje con el registro E. Por eso hay que hacer el decremento y salto usando dos instrucciones separadas.

Si intentas alterar estos dos valores, verás cómo el valor cargado en el registro E modifica considerablemente la anchura de las bandas, mientras que el valor contenido en el registro B sólo las altera ligeramente.

COMPROBACION DE «BREAK»

Tiene que haber algún punto desde el que puedas escapar de esta rutina. Para poder hacerlo sin necesidad de apagar tu ordenador, tienes que incluir una rutina de escape.

La que se utiliza aquí realiza una comprobación de si se ha pulsado o no la tecla BREAK. Esto se hace utilizando la instrucción **in**.

Pero previamente la instrucción **ld d,a** carga el contenido del acumulador en el registro D. En seguida vamos a utilizar el acumulador para otra cosa, mientras que el registro D no está ha-

P y R

¿Cuántos *ports* hay en el sistema?

Teóricamente son posibles 64 K *ports*, ya que éste es el mayor número que puede ser direccionado con 16 bits. Pero en la práctica sólo se usan unos cuantos de los posibles *ports*. Cuando tu **Spectrum** está en una configuración de *hardware* ordinaria, sólo se utiliza un *port*, el 254.

Ya has visto cómo al introducir diferentes parámetros en el comando **in**, se dirige el *port* hacia diferentes zonas del teclado. Análogamente, diferentes bits del comando **out** del código máquina, permiten controlar a dispositivos periféricos distintos.

No obstante, si tu **Spectrum** está conectado a otros dispositivos no estándar, se pueden utilizar otros *ports*. Un ejemplo sencillo es el caso en que se utiliza un *joystick* **Kempston** conectado al **Spectrum**; para acceder a él se utiliza el *port* 31.

Si tienes buena mano para la electrónica, podrías conectar tu **Spectrum** para controlar tu sistema de calefacción central o tu sintetizador electrónico por medio de diferentes *ports*.

ciendo nada, por lo que podemos utilizarlo como almacenamiento temporal.

Ahora cargamos el acumulador con el valor hexadecimal 7F. Este número especifica la esquina inferior derecha del teclado. La instrucción **in a,254** toma el conjunto de bits que definen el estado de la parte inferior derecha del teclado y lo almacena en el acumulador.

El bit 0 de dicho conjunto es el que define el estado de la tecla BREAK. Si dicha tecla no está siendo pulsada, el valor del bit 0 es 1, pero si está siendo pulsada, el valor de dicho bit es 0.

Una manera rápida de comprobar esto es aplicar una rotación al bit 0, que es el del extremo derecho del registro, hasta llegar al indicador de acarreo (*carry flag*), y a continuación examinar el estado de dicho indicador. La instrucción **rra** realiza la rotación, y la **rts nc** ejecuta el retorno al BASIC cuando no hay acarreo. Así, cuando se ha pulsado BREAK y el bit 0 pasa de 1 a 0, la instrucción **rts** fuerza la salida de la rutina. En caso contrario ésta continúa. (Otros ensambladores del **Spectrum** utilizan **ret** para los retornos condicionales).

CERRANDO LOS BUCLES

Ahora que ya tienes una manera de escapar de la rutina, se puede restaurar el valor del acumulador con la instrucción **ld a,d**; esta instrucción se limita a cargar de nuevo en A el valor que habíamos almacenado temporalmente en D.

Seguidamente se incrementa A para especificar el siguiente color. Este se compara con el número 7 por medio de **cmp 7**. El 7 es el número con el que se especifica el blanco, que es el color de la pantalla.

Si el número contenido en el acumulador no es el que especifica el color blanco, la instrucción **jrnz,bucle** fuerza un salto para salir al color de la siguiente banda. Cuando este bucle ha realizado una cuenta completa desde 0 hasta 7, el microprocesador pasa a la instrucción **jr repe** que le envía de nuevo al principio para que espere la siguiente exploración de pantalla.

Habrás visto que no tienes que preocuparte de que se produzcan accidentalmente sonidos al cambiar el color del borde de pantalla. Ello se debe a que esta rutina no afecta al bit 4 que es el que mueve el altavoz. El número más alto al que aquí se da salida con la instrucción **out** es el 7, mientras que el número más bajo al que afectaría el bit 4 es el 16.

EFECTOS SONOROS

Ha llegado el momento de abordar la realización de efectos sonoros más complejos con ayuda de la instrucción **out**. La siguiente rutina da como resultado un disparo de rayos laser del más puro estilo de la guerra de las galaxias, combinando un sonido de altura creciente con otro de altura decreciente.

```
        org 64600
        ld a,(23624)
        rrca
        rrca
        rrca
        ld b,0
bucle   push bc
        xor $10
        out 254,a
        push af
        xor a
        sub b
        ld b,a
        pop af
pausaa  nop
        djnz pausaa
        xor $10
        out 254,a
        pop bc
        push bc
pausab  nop
        djnz pausa b
        pop bc
        djnz bucle
        ret
```

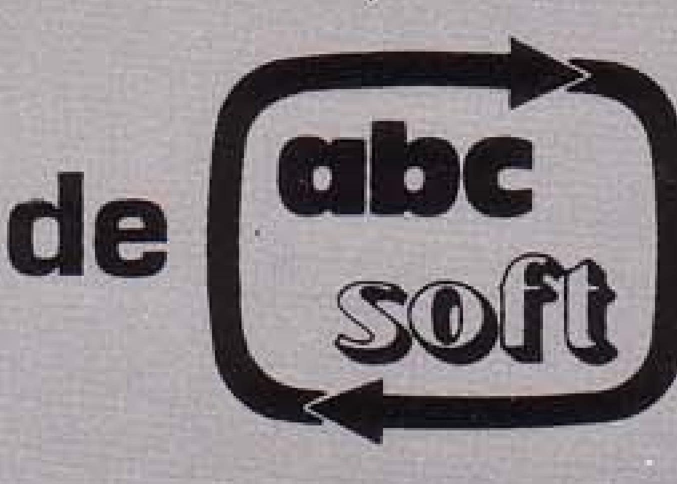
Las cuatro primeras instrucciones de este programa son exactamente las mismas que las cuatro primeras del programa de sonidos. Recuerda que preservan el color del borde de pantalla.

Las dos instrucciones siguientes —que inicializan los dos contadores, uno en el registro B y el otro transferido desde el registro B al *stack*— son también iguales. Como también lo son las dos siguientes que realizan una operación OR exclusiva del bit 4 con 16, y sacan el resultado por el *port* 254; sin embargo esta vez el **xor** va precedido de $10 que es 16 en hexadecimal. Así se genera el sonido.

Pero ahora el contenido del acumulador es cargado en el *stack* con la instrucción **push**, junto con el contenido del registro de indicadores. Los registros tienen que ser transferidos por parejas. La instrucción **xor a** borra el



de





Estos programas son los más vendidos en el Reino Unido, según las encuestas de ventas realizadas por GALLUP/MICROSCOPE, en las semanas del 14 de Abril al 16 de Mayo de 1986.

SPECTRUM

	TITULO	EDITOR	P.V.P.R.
1 ( ) [3]	BATMAN	OCEAN	
2 (1) [8]	GREEN BERET	IMAGINE	2.100
3 ( ) [6]	STARSTRIKE 2	REALTIME	2.575
4 ( ) [4]	HEAVY ON THE MAGIC	GARGOYLE	
5 (3) [6]	BOMB JACK	ELITE	2.575
6 (2) [8]	WAY OF THE TIGER	GREMLIN	2.100
7 (8) [5]	V	OCEAN	2.300
8 (5) [7]	INCREDIBLE SHRINK	MASTERTRONIC	750
9 (4) [7]	TURBO SPRIT	DURELL	2.100
10 ( ) [2]	CYBERUN	ULTIMATE	

COMMODORE 64

	TITULO	EDITOR	P.V.P.R.
1 ( ) [3]	THRUST	FIREBIRD	1.200
2 ( ) [3]	INTERNATIONAL KARATE	SYSTEM 3	2.350
3 ( ) [3]	SPINDIZZY	ELECTRIC DREAMS	2.450
4 ( ) [5]	BOMB JACK	ELITE	2.575
5 ( ) [4]	PSI-5 TRADING COMPANY	ACCOLADE	2.300
6 ( ) [4]	SPELLBOUND	MASTERTRONIC	1.100
7 ( ) [4]	THEY SOLD A MILL. 2	HIT SQUAD	
8 (1) [8]	URIDIUM	HEWSON CONS.	2.500
9 (4) [5]	V	OCEAN	2.300
10 ( ) [3]	FORMULA ONE SIM.	MASTERTRONIC	750

AMSTRAD

	TITULO	EDITOR	P.V.P.R.
1 (4) [8]	LAST V8	MASTERTRONIC	1.100
2 (3) [8]	SPINDIZZY	ELECTRIC DREAMS	2.450
3 ( ) [6]	GET DEXTER	P S S	2.600
4 ( ) [5]	THEY SOLD A MILL 2	HIT SQUAD	
5 ( ) [4]	COMMANDO	ELITE	2.575
6 (7) [6]	INTO OBLIVION	MASTERTRONIC	750
7 (5) [9]	COMPUTER HITS V2	BEAU JOLLY	3.400
8 (1) [7]	WAY OF THE TIGER	GREMLIMS	2.300
9 ( ) [3]	BATMAN	OCEAN	
10 ( ) [3]	TURBO SPRIT	DURELL	

COMMODORE 16

	TITULO	EDITOR	P.V.P.R.
1 ( ) [3]	KIK START	MASTERTRONIC	750
2 (1) [13]	MR PUNIVERSE	MASTERTRONIC	750
3 ( ) [7]	BANDITS AT ZERO	MASTERTRONIC	1.100
4 (4) [19]	FORMULA ONE SIM.	MASTERTRONIC	750
5 (3) [15]	BIG MAC	MASTERTRONIC	750

MSX

	TITULO	EDITOR	P.V.P.R.
1 (3) [6]	WAY OF THE TIGER	GREMLIMS	2.300
2 (2) [7]	FINDERS KEEPERS	MASTERTRONIC	750
3 (1) [8]	FORMULA ONE SIM.	MASTERTRONIC	750
4 (4) [5]	CHILLER	MASTERTRONIC	750
5 (5) [2]	TURMOIL	BUG BYTE	1.200

( ) Posición en mes anterior, [ ] n.º de semanas entre los más vendidos.

P.V.P.R.: Precio de venta recomendado, incluido IVA

* Encuéntralos en la división

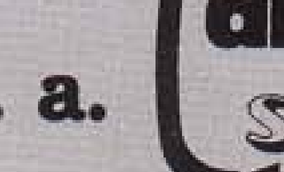


de GALERIAS

* Y en comercios de microinformática

* Es una promoción de:

abc analog, s. a.



Santa Cruz de Marcenado, 31 (3.º 14) 28015 MADRID

Tfnos: (91) 248 82 13 - 242 50 59
Télex. 44561 BABC E

HACIENDO MUSICA CON EL SPECTRUM

Se puede hacer música con el **Spectrum** utilizando la instrucción **out** del código máquina. Pero ajustar el bucle pausa al tono requerido («afinar» el bucle) puede resultar laborioso.

Sería mucho más fácil escribir una rutina a la que poder darle dos parámetros, relacionados con la altura del sonido y su duración. Afortunadamente, esto ya lo tienes hecho en ROM. Es la rutina BEEP y comienza en la dirección 03F8. Hay además otra subrutina adicional llamada BEEPER, que comienza en 03B5 y utiliza el valor de HL para controlar el tono y el valor de DE para controlar la duración.

El valor del tono que tienes que cargar en HL está determinado por 437500/f-30.125 donde f es la frecuencia de la nota. Para obtener el número que tienes que introducir en DE, multiplica la frecuencia por la duración de la nota. Al utilizar este método no tienes la limitación de 10 segundos para la duración de la nota.

Si llamas a la rutina BEEP directamente, tienes que pasar al *stack* mediante una instrucción **push** los mismos valores de duración y tono utilizados con la subrutina BEEP del BASIC.

acumulador, mientras que la **sub b** separa el contenido no nulo de B —el contenido de a— y almacena el resultado en A. Esto invierte realmente el contenido del registro B. Cuando B es 255, el resultado de la diferencia es 1; cuando B es 1, el resultado es 255.

La instrucción **ld b,a** carga en B el resultado de la resta. El contenido del acumulador, que estaba almacenado en el *stack*, se transfiere nuevamente al acumulador con la instrucción **pop**.

Seguidamente se produce un bucle de pausa que realiza una cuenta atrás del contenido del registro B; recuerda que **djnz** decrementa y salta si el resultado no es cero. El bit 4 es modificado y sacado hacia el altavoz.

El valor inicial del registro B se restaura haciendo un **pop** del *stack*. Dicho valor será necesario después, por lo que después de copiarlo en el registro B se le hace un nuevo **push** al *stack*; así pues queda almacenado en dos sitios: en el registro B y en el *stack*.

El siguiente bucle de pausa opera sobre este valor inicial de B, con el que hace un nuevo **pop** en el *stack*. Hay que hacer esto después de cada instruccción **djnz** a fin de restaurar el valor del registro B. Cuando el micro sale de un bucle **djnz** su valor debe ser cero.

A continuación se decrementa el valor inicial del registro B. Si dicho valor no es cero, el microprocesador vuelve al bucle y se ejecuta de nuevo toda la rutina con un nuevo valor de B. Al llegar al paso número 256, en el que B se decrementa finalmente a cero, el micro salta a la instrucción **ret**, regresando al BASIC.

Como puedes ver, el bucle pausaa se ejecuta 256 veces en la primera pasada, una vez en la segunda pasada y una vez más cada vez que se ejecuta el bucle principal.

SONIDOS NATURALES

Los sonidos generados por los ordenadores resultan normalmente muy sintéticos debido a que son demasiado puros. Los instrumentos musicales y otros dispositivos productores de sonido en el mundo exterior tienden a comportarse de una forma bastante aleatoria en la forma en que producen el sonido. Pero es precisamente esta aleatoriedad lo que los hace agradables al oído humano. Un instrumento no sólo emite el sonido fundamental, correspondiente a un tono puro, sino también los armónicos del mismo.

No obstante, existe la posibilidad de añadirle un factor de aleatoriedad al sonido producido por el **Spectrum**. La memoria RAM correspondiente a las direcciones entre 16384 y 32767 está instalada en *chips* que frecuentemente se ven interrumpidos por un «*array* lógico» que se ocupa de la imagen del televisor y realiza otras funciones propias del *hardware*.

Normalmente estas interrupciones son tan rápidas —típicamente un par de microsegundos— que ni se notan. Pero los sonidos tienen una dependencia tan estricta del tiempo que el oído puede detectar incluso las más ligeras variaciones. Por eso, si sitúas tus programas de efectos de sonido en esta zona de la memoria, conseguirás una sensación aleatoria más natural, y parecerán ligeramente más largos.

El *buffer* de la impresora se encuentra entre las direcciones 23296 y 23552. Puedes pués utilizar como origen 23296. Y como el *buffer* de la impresora está protegido contra sobreescritura por parte del BASIC, no tienes que hacer un CLEAR antes del ensamblado.



LO MEJOR PARA SPECTRUM







EL RETO A TU DESTREZA EN TRES DIMENSIONES

- Con SPINDIZZY te trasportaras a un enigmático mundo de fantasía.
- Sus casi 400 pantallas se convierten en fuente inagotable de diversión.
- Los mejores gráficos tridimensionales que un AMSTRAD puede generar en un juego de este tipo.
- Tu mismo puedes alterar el trazado de las pantallas, situándote cada vez ante un nuevo desafío: rampas, ascensores, trampas, etc.

TAMBIEN DISPONIBLE PARA AMSTRAD Y COMMODORE

Electric Dreams SOFTWARE

EN TIENDAS ESPECIALIZADAS Y GRANDES ALMACENES, O DIRECTAMENTE POR CORREO O TELEFONO A: **PROEIN, S.A.**
Velázquez, 10 - 28001 Madrid Tels. (91) 276 22 08/09

MANEJANDO GRAFICOS CON EL JOYSTICK

Este programa permite manejar rectas, cuadrados y círculos de cualquier tamaño y posición dentro de la pantalla.

Para aquellos que aún no se hayan decidido por un *joystick* determinado, hemos previsto la utilización del teclado del **ZX-Spectrum**. El resultado es el mismo, aún cuando evidentemente resulte menos manejable.

En la parte inferior de la pantalla queda reflejada en todo momento la posición del puntero. Las facilidades que ofrece el programa quedan agrupadas en dos menús, los cuales pueden hacerse patentes, alternativamente, en la esquina inferior derecha de la pantalla. Todos los movimientos, incluida la selección de opciones se realiza mediante el *joystick*.

Todo el programa está realizado en BASIC, a pesar de lo cual tiene una agilidad suficiente. Se ha procurado resaltar su aspecto didáctico y para ello se han hecho agrupaciones lógicas con abundantes indicaciones. En definitiva, se ha sacrificado algo de rapidez de ejecución en aras de una mejor comprensión.

El *joystick* utilizado dispone de cuatro posiciones básicas (adelante, atrás, derecha e izquierda), más la de reposo y un gatillo disparador. El gatillo tiene tres variantes en cuanto a sensibilidad (dos pulsadores en la empuñadura y uno en la base) pero su acción es la misma. La señalización de las cuatro posiciones básicas es distinta cuando se oprime simultáneamente el disparador.

La detección de la posición del *joystick* se obtiene leyendo una variable (en nuestro caso hemos utilizado '*stick*') el valor de la puerta 31 mediante 'IN 31' (instr. 220). Los valores obtenidos son los siguientes:

Reposo=0; Adelante=8; Atrás=4; Derecha=1; Izquierda=2 Disparo=16

Siempre que se actúe con el disparador se suma 16 al valor correspondiente.

En el caso del teclado hemos utilizado 0 para el gatillo, las flechas (5,6,7,8) y las letras T, Y, U, I para señalizar flechas + gatillo. En este caso la detección de la tecla pulsada se hace mediante INKEY$ (instr. 8000 / 8500)

El desplazamiento dentro de las opciones del menú en servicio se consigue, circularmente, mediante el gatillo (o la tecla '0' en su caso). Para cambiar de menú basta con empujar la palanca hacia delante cuando se está en «menú».

Resulta aconsejable ir guardando pantallas a medida que progresamos en el dibujo para evitar perderlo todo ante una mala maniobra. Puede ser conveniente almacenar una misma pantalla con diversos nombres a fin de poder hacer tanteos y complementar datos con otros programas.

A quien tenga interés, estudiando el programa no le será difícil añadir nuevos menús para incluir opciones de rotulación, coloreado, etc.

Igualmente, compactando ciertas instrucciones, se podrá conseguir algo más de velocidad, aunque lo aconsejable, en este caso sería acudir al código máquina.

Está previsto que en el caso de intentar el dibujo de una circunferencia que salga fuera de la pantalla nos de un mensaje de aviso para rectificar los puntos de referencia. En el caso de que nos empeñemos en salir de la pantalla, moviendo el *joystick*, en cualquiera de las cuatro direcciones, el cursor quedará bloqueado momentáneamente. Mediante GO TO 9000 ó GO TO 9999 se facilita la labor de almacenar y copiar el programa en *cassette* o *diskette*.





```
10 REM programa para Joystick
15 REM
20 LET borde=7: LET papel=7:
   LET tinta=0
30 BORDER borde: INK tinta:
   PAPER papel: CLS
40 CLS : PRINT AT 2,2;
   "EJEMPLO MANEJO DE
   JOYSTICK";AT 4,4;"REVISTA
   INPUT SINCLAIR";AT 6,8;
   "MOLISOFT 1986"
60 PRINT AT 15,0;"1 -
   JOYSTICK DISCOVERY 1";AT
   17,0;"2 - TECLADO ZX-
   SPECTRUM 48K";AT 21,0;
   "OPCION ?"
70 LET t$=INKEY$: IF t$=""
   THEN GO TO 70
80 IF t$<>"1" AND t$<>"2"
   THEN GO TO 70
90 CLS : PLOT 0,0: DRAW 0,175
   : DRAW 255,0: DRAW 0,-175:
   DRAW -255,0
99 REM
100 GO SUB 1200: REM reset
    flashes
101 REM
110 LET x=128: LET x0=128:
    LET y=88: LET y0=88: LET
    xa0=0: LET xa=0: LET xb0
    =0: LET xb=0: LET ya0=0:
    LET ya=0: LET yb0=0: LET
    yb=0: LET k=1: LET m=1:
    LET ok=0
120 GO SUB 3500: GO SUB 1000
    : GO SUB 1100: PLOT x0,y0
150 IF t$="1" THEN GO TO 200
160 GO TO 8000
169 REM
200 REM EXPLORACION JOYSTICK
201 REM
210 FORMAT "j";1
220 LET stick=IN 31: IF stick
    =0 THEN GO TO 220
230 IF stick=16 THEN LET k=k
    +1: LET k=FN k(): GO SUB
    1200: GO SUB 3500: GO
    SUB 1100
237 REM
238 REM MENU1
239 REM
300 IF m<>1 THEN GO TO 400
301 IF stick=8 AND (f1=1 OR
    f2=1 OR f3=1) THEN LET
    y=y0+(y0<174): GO TO 310
302 IF stick=24 AND (f1=1 OR
    f2=1 OR f3=1) THEN LET
```

Menu 1	Posicion	Descripcion
EXploracion	Adelante	Movimiento del puntero hacia arriba (pixel a pixel) para situarle en un lugar determinado o para conocer las coordenadas de una posicion determanada. Sirve tambien para borrar pixel a pixel.
	Atras	idem hacia abajo
	Derecha	idem hacia la derecha
	Izquierda	idem hacia la izquierda
	Adel+gat.	idem adelante pero de 10 en 10 pixels
	Atr.+gat.	idem atras pero de 10 en 10 pixels
	Der.+gat.	idem derecha pero de 10 en 10 pixels
	Izq.+gat.	idem izquierda pero de 10 en 10 pixels
INvertir	A/A/D/I	Idem EXploracion pero invirtiendo los pixel al paso del puntero.
	id.+gat.	Idem INvertir pero afectando solo a cada 10 pixels.
TRazar	A/A/D/I	trazado continuo, pixel a pixel
	id.+gat.	trazado cada 10 pixels.
Ma	adelante	Dejar marcado punto 'a'
Mb	adelante	Dejar marcado punto 'b'

```
   y=y0+10*(y0<165): GO TO
   310
303 IF stick=4 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET y
   =y0-(y0>1): GO TO 310
304 IF stick=4 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET y
   =y0-(y0>1): GO TO 310
305 IF stick=20 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET y
   =y0-10*(y0>10): GO TO 310
306 IF stick=1 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET x
   =x0+(x0<254): GO TO 310
307 IF stick=17 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET x
   =x0+10*(x0<245): GO TO
   310
308 IF stick=2 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET x
   =x0-(x0>1): GO TO 310
309 IF stick=18 AND (f1=1 OR
   f2=1 OR f3=1) THEN LET x
   =x0-10*(x0>10)
310 GO SUB 1000
315 IF f1=1 THEN PLOT OVER 1
   ;x0,y0: PLOT x,y: GO TO
   800
320 IF f2=1 THEN PLOT OVER 1
   ;x,y: GO TO 800
325 IF f3=1 THEN PLOT x,y:
   GO TO 800
330 IF f4=1 AND stick=8 THEN
   BEEP .5,10: LET xa=x:
   LET ya=y: GO TO 800
340 IF f5=1 AND stick=8 THEN
   BEEP .5,10: LET xb=x:
   LET yb=y: GO TO 800
348 REM MENU 2
349 REM
400 IF m<>2 THEN GO TO 800
410 IF f1=1 AND stick=8 AND
   ok=1 THEN CIRCLE xa,ya,r
   : GO TO 800
412 IF f1=1 AND stick=4 AND
   ok=1 THEN CIRCLE OVER 1;
   xa,ya,r: GO TO 800
414 IF f1=1 AND (stick=4 OR
   stick=8) AND ok=0 THEN
   PRINT #0;AT 0,0; FLASH
   1;"[5*ESPACIO]valores
   incorrectos[7*ESPACIO]":
   BEEP 2,10: PRINT #0;AT
   0,0;"[31*ESPACIO]: GO
   SUB 1000: GO SUB 1100:
   GO TO 800
420 IF f2=1 AND stick=8 THEN
   PLOT xa,ya: DRAW d0,0:
   DRAW 0,h0: DRAW -d0,0:
   DRAW 0,-h0: GO TO 800
422 IF f2=1 AND stick=4 THEN
   PLOT xa,ya: DRAW OVER 1;
   d0,0: DRAW OVER 1;0,h0:
   DRAW OVER 1;-d0,0: DRAW
   OVER 1;0,-h0: GO TO 800
430 IF f3=1 AND stick=8 THEN
   PLOT xa,ya: DRAW d0,h0:
   GO TO 800
432 IF f3=1 AND stick=4 THEN
   PLOT xa,ya: DRAW OVER 1;
   d0,h0: GO TO 800
440 IF (f4=1 OR f5=1) AND
   stick=8 AND t$="1" THEN
   FORMAT "j";0: GO SUB
   4010
442 IF (f4=1 OR f5=1) AND
   stick=8 AND t$="2" THEN
   GO SUB 4010
443 IF f4=1 AND stick=8 AND
   t$="1" THEN SAVE *"m";1;
   p$SCREEN$ : FORMAT "j";1
   : GO TO 4100
445 IF f4=1 AND stick=8 AND
   t$="2" THEN SAVE
   p$SCREEN$ : GO TO 4100
448 IF f5=1 AND stick=8 AND
   t$="1" THEN LOAD *"m";1;
   p$SCREEN$ : FORMAT "j";1
   : GO TO 4100
450 IF f5=1 AND stick=8 AND
   t$="2" THEN LOAD p$: GO
   TO 4100
800 IF f6=1 AND stick=8 THEN
   BEEP .5,10: LET m=m+1:
   LET m=FN m(): GO SUB
   1100
900 LET x0=x: LET y0=y
910 LET d0=(xb-xa): LET d=ABS
   d0: LET h0=(yb-ya): LET
```

GANADORES DE LOS MEJORES DE INPUT SINCLAIR

En el sorteo correspondiente al número 9 entre quienes escribisteis mandando vuestros votos a LOS MEJORES DE INPUT han resultado ganadores:

NOMBRE	LOCALIDAD	JUEGO ELEGIDO
José Cañas Landin	Vigo (Pontevedra)	Movie
Federico Lechner Loubet	Madrid	Evil Crown
Julián Nicolao Echevarne	Pamplona	Winter Games
M.ª Manuela Sagarruy Galindo	Madrid	Movie
Juan José López Fontanet	Chiva (Valencia)	Shadow of Unicorn
Andrés Mondéjar Siret	La Eliana (Valencia)	Basketball International
Fernando García Esteban	Madrid	Back to the Future
Juan Grau Burgués	Lérida	Skyfox
Silvia Delgado Martín	Madrid	Movie
Emilio De Diego Magdaleno	Madrid	Skyfox

Menu 1	Posicion	Descripcion
------	--------	--
CIrculo	adelante	Trazado de circulo de centro en 'a' que pasa por 'b'
--		
	atras	idem borrado
REctang.	adelante	Trazado de un cuadrado cuya diagonal esta definida por 'a' y 'b'
--		
	atras	idem borrado
LInea	adelante	trazado de recta 'a'/'b'
--		
	atras	idem borrado
SAve	adelante	archivo en disco de la pantalla
--		
LOad	adelante	carga de una pantalla desde disco
--		

```
    h=ABS h0: LET r=FN r()
920 LET ok=(ya+r)<175 AND
    (ya-r)>0 AND (xa+r)<255
    AND (xa-r)>0
930 PLOT OVER 1;xa0,ya0:
    PLOT OVER 1;xb0,yb0:
    PLOT xa,ya: PLOT xb,yb:
    LET xa0=xa: LET xb0=xb:
    LET ya0=ya: LET yb0=yb
990 IF t$="1" THEN GO TO 220
998 GO TO 160
999 REM
1000 REM INFORMACION DEL
     PUNTERO
1001 REM
1010 PRINT #0;AT 0,0;"x=[3*
     ESPACIO]";AT 0,2;x;AT 0
     ,6;"y=[3*ESPACIO]";AT
     0,8;y
1030 RETURN
1039 REM
1100 REM INFORMACION FLASHES
1109 REM menu 1
1110 REM
1111 IF m=1 THEN PRINT #0;AT
     0,12; FLASH f1;"Ex";AT
     0,15; FLASH f2;"In";AT
     0,18; FLASH f3;"Tr";AT
     0,21; FLASH f4;"Ma";AT
     0,24; FLASH f5;"Mb";AT
     0,27; FLASH f6;"menu"
1118 REM
1119 REM menu 2
1120 REM
1121 IF m=2 THEN PRINT #0;AT
     0,12; FLASH f1;"Ci";AT
     0,15; FLASH f2;"Re";AT
     0,18; FLASH f3;"Li";AT
     0,21; FLASH f4;"Sa";AT
     0,24; FLASH f5;"Lo";AT
     0,27; FLASH f6;"menu"
1130 RETURN
1139 REM
1200 REM REPOS. VALORES FLASH
1201 REM
1210 LET f1=0: LET f2=0: LET
     f3=0: LET f4=0: LET f5=
     0: LET f6=0: RETURN
1219 REM
2000 REM funciones definidas
2001 REM
2010 DEF FN k()=k-6*(INT (k/
     7))
2020 DEF FN m()=m-2*(m=3)
2030 DEF FN r()=INT (SQR ((d
     ^2)+(h^2)))+1
3499 REM
3500 REM DEF. DE FLASHES
3501 REM
3510 IF k=1 THEN LET f1=1
3520 IF k=2 THEN LET f2=1
3530 IF k=3 THEN LET f3=1
3540 IF k=4 THEN LET f4=1
3550 IF k=5 THEN LET f5=1
3560 IF k=6 THEN LET f6=1
3600 RETURN
3699 REM
4010 BEEP .5,10: INPUT FLASH
     1;"nombre?+ENTER(max.7c
     )";p$: IF LEN p$>7 THEN
     GO TO 4010
4020 RETURN
4100 GO SUB 1000: GO SUB
     1100: GO TO 800
8000 REM PROGRAMACION TECLADO
8001 REM
8010 LET h$=INKEY$: IF h$=""
     THEN GO TO 8010
8020 IF h$="5" THEN LET
     stick=2: GO TO 8500
8030 IF h$="t" THEN LET
     stick=18: GO TO 8500
8040 IF h$="6" THEN LET
     stick=4: GO TO 8500
8050 IF h$="y" THEN LET
     stick=20: GO TO 8500
8060 IF h$="7" THEN LET
     stick=8: GO TO 8500
8070 IF h$="u" THEN LET
     stick=24: GO TO 8500
8080 IF h$="8" THEN LET
     stick=1: GO TO 8500
8090 IF h$="i" THEN LET
     stick=17: GO TO 8500
8100 IF h$="0" THEN LET
     stick=16: GO TO 8500
8400 GO TO 8000
8500 GO TO 230
8999 REM
9000 SAVE "stick" LINE 10:
     GO TO 10: REM cassette
9500 REM
9999 SAVE *"m";1;"stick"
     LINE 10: GO TO 10: REM
     discovery 1
```

LOS DINAMIC CAMBIAN DE MANSION



En el piso ventinueve de la conocida Torre de Madrid, edificio que alberga las oficinas de importantes firmas españolas y extranjeras, acaba de abrir la suya una nueva empresa española dedicada a la comercialización de *software*. Su nombre Microdigital S.A.

En los actuales momentos son numerosas las empresas que se dedican en nuestro país a la distribución de *software* de importación, y de vez en cuando, surge alguna nueva. Sin embargo, **Microdigital** es diferente. Por un lado porque se dedica a la creación y distribucción de *software* español, porque en lugar de importar exporta,



pero sobre todo, porque tras ese nombre de gran compañía está otro que nos resulta mucho más familiar, a todos los aficionados a la microinformática. Nos estamos refiriendo a **Dinamic**, la casa de *soft* que ya hace dos años tenía ecos de mansión.

Para conocer mejor los proyectos de esta nueva empresa y sobre todo para charlar de la historia de **Dinamic** —cuya evolución a llevado a sus protagonistas de los estudios preuniversitarios a mantener relaciones comerciales en varios paises europeos— hemos ido a visitarles y allí hemos conversado largo y tendido con **Pablo**, **Victor**, **Nacho**, **Jesús** y **Santiago**, que nos contaron toda su historia.

DE LA MANSION AL RASCACIELOS

Pablo, **Victor** y **Nacho Ruiz**, son tres hermanos que allá por 1984 alternaban sus obligaciones estudiantiles con una afición que entonces empezaba a difundirse: teclear sus propios programas. Los primeros pasos en programación los habían dado un año antes, como tantos otros, con el **Sinclair ZX-81**.

Para la primavera del 84, **Victor** y **Nacho** ya habían desarrollado unos programas. Pero fue **Pablo** el que pensó que aquello era bueno y podía venderse.

Preguntaron cuánto costaba una página de publicidad en **ZX**, la única revista que existía por aquel entonces. Como estaban ahorrando para comprarse un **QL** disponían de unas 80.000 pesetas. que distribuyeron de la siguiente forma: 40.000 para insertar media página de publicidad; 17.000 para la impresión de carátulas; 11.000 para cintas, y el resto para sobres y sellos.

Dos años después se rien recordando aquellos días emocionantes. «**Santiago** había hecho un dibujo estupendo para la publicidad, y los demás contribuimos poniendo los letreros con **Letrasset**, lo hicimos todo la noche antes de entregarlo».

- Insertastéis la publicidad, y ¿qué pasó?

- «Pues que nos empezaron a llamar como locos. Recibíamos los pedidos y los mandábamos por correo. Lo hacíamos todo: grabábamos los programas uno a uno, pasándolos directamente desde el **Spectrum** al *cassette*. Por esto proclamábamos que nuestros programas tenían el mejor control de calidad, ya que eran comprobados uno a uno...»

Por las tardes copiaban los programas que tenían que mandar al día siguiente. «Lo malo fue cuando una tienda nos hizo el primer pedido importante, nos pasamos toda la noche en blanco».

Vendiendo los programas a 1.000 pesetas el éxito fue enorme. Como faltaban manos, **Jesús** dejó su trabajo en una compañía de seguros y se incorporó a la cadena de empaquetado y envío por correo.

Desde el primer momento **Pablo** había organizado la empresa de forma que **Victor** y **Nacho** fueran los programadores, y **Santiago** hiciera los dibujos mientras él, actuaba como director.

- ¿Cómo trabajáis en los programas?

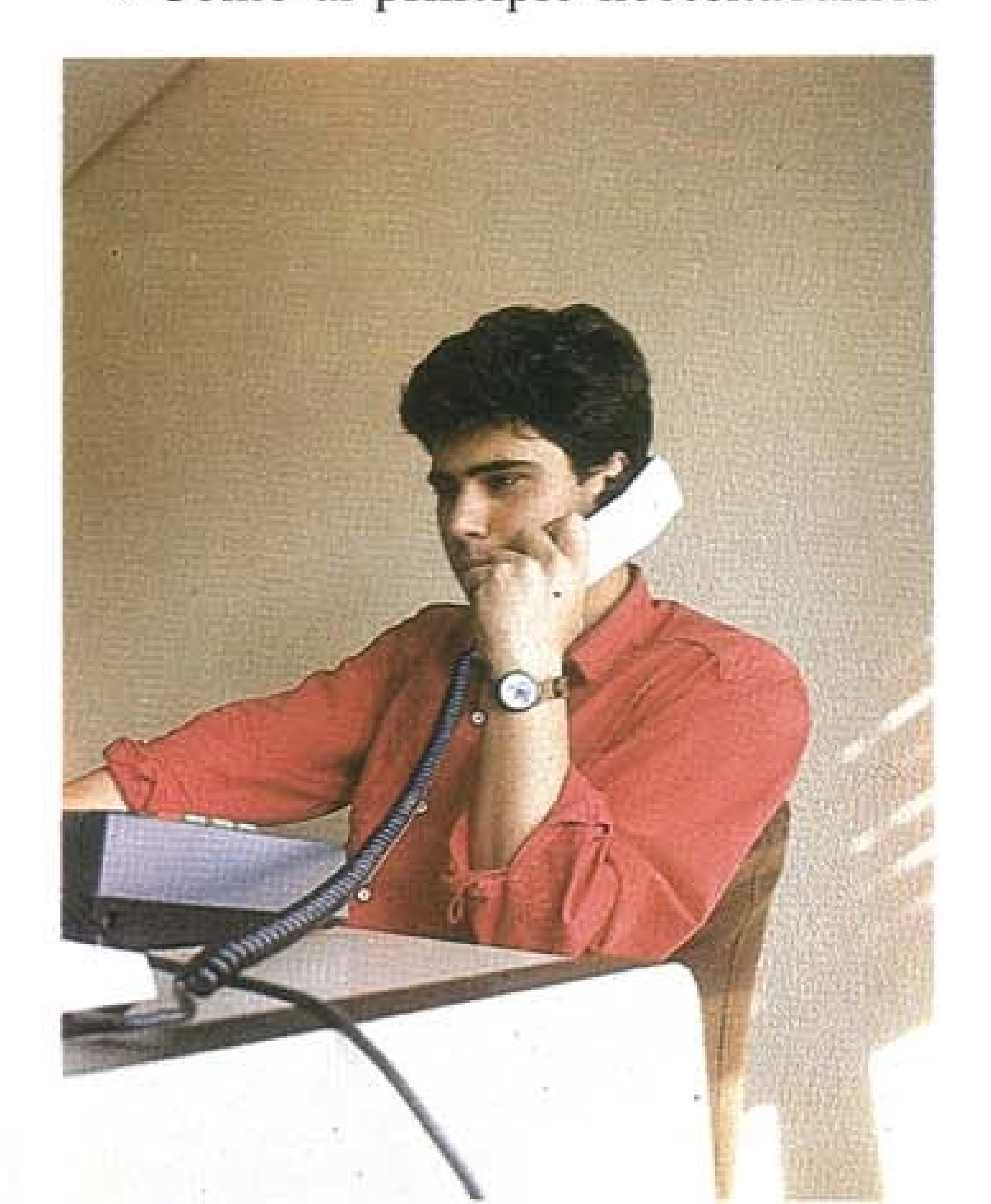
- A partir de una idea, es el programador, ya sea **Victor** o **Nacho**, el que lleva la voz cantante. Los demás apuntamos ideas que el programador retoma o no. A veces hasta discutimos sobre la paternidad de la idea, pero siempre son ellos dos los que deciden lo que se incorpora o no.

El proceso es siempre más o menos el mismo. A alguien se le ocurre una idea y entonces se diseñan los gráficos y se «mapea» el juego. Como hay dos programadores hemos ido haciendo los juegos de dos en dos, y por eso más de una vez se ha adaptado a un programa alguna rutina, idea o personaje que en principio iba a ir en el otro juego».

En poco tiempo, el nombre de **Dinamic** fue extendiéndose hasta ser el fabricante más de entre los usuarios de **Spectrum**. Por cierto que acerca de esto hay que comentar por qué acuñaron el nombre de **Mansión Dinamic**. «El caso es que nos fijamos en que los ingleses incorporaban el término *house* (casa) en muchos de sus nombres comerciales, y pensamos hacer lo mismo ya que sonaba bien. Pero «casa» no acababa de convencernos, y aunque barajamos otras posibilidades, como castillo o palacio, nos decidimos por lo de mansión».

Aunque el nombre puede parecer un tanto rimbombante, no deja de ser toda una declaración de intenciones que nos habla de la ilusión con que emprendieron su empresa, y de la que es un buen ejemplo la siguiente anécdota de lo que ellos llaman la «primera capitalización».

« Como al principio necesitabamos





dinero y no teníamos más que lo que iba llegando de la venta por correo, nos pusimos a pensar en la forma de conseguirlo y a **Victor** se le ocurrió la idea de escribir un cuento y presentarlo a un premio literario». Dicho y hecho. Todos se pusieron a la obra y consiguieron ganar un premio de 100.000 pesetas. Sobran los comentarios.

LOS PRIMEROS COLABORADORES

Después del **Saimazon** y del **Mapis Snatch** hicieron **Babaliba** y **Villaolimpic** (los programas siempre aparecían de dos en dos) y las disponibilidades de la empresa fueron incrementándose y sus medios técnicos fueron perfeccionándose.

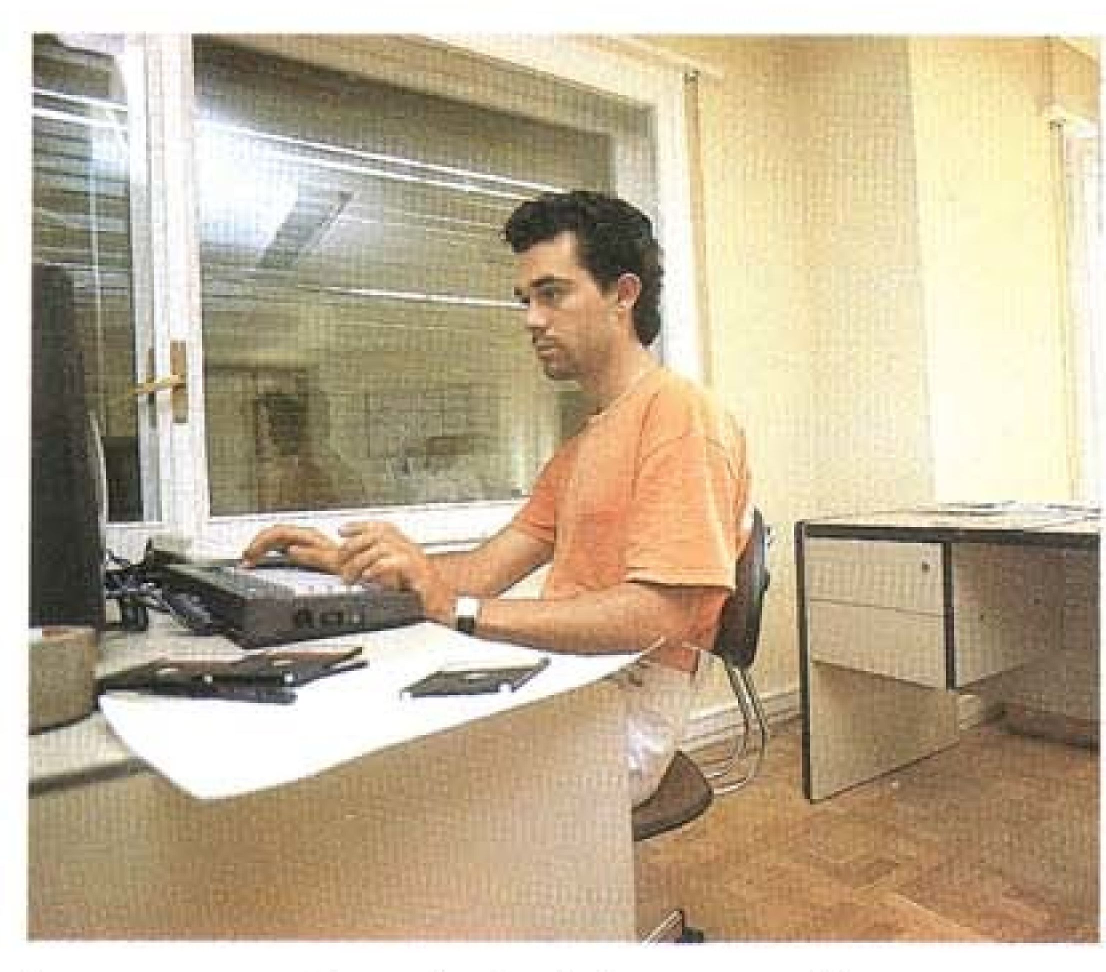
«El primer programa lo hicimos con el **Spectrum** conectado a un televisor en blanco y negro, y a un *cassette*; después compramos un *microdrive* pero utilizabamos el televisor en color de nuestros padres. No fue hasta hacer el **Profanation** cuando compramos una unidad de disco y un monitor en color».

Pero por otra parte la popularidad de **Dinamic** hizo que muchos programadores hasta ese momento «aficionados» les enviaran sus programas con la intención de comercializarlos, iniciándose así una segunda etapa de la empresa en la que ésta comercializa el trabajo de *Free Lances*, es decir de personas que al trabajar como autónomos, no forman parte en sentido estricto de la empresa, pero venden un programa bajo su sello.

«El primero de ellos fue el **Rocky**. Cuando lo vimos nos gustó pero fue necesario hacerle bastantes modificaciones. Desde entonces el proceso se ha repetido varias veces. Vemos un programa y si nos parece bien comunicamos a sus autores los cambios que han de hacer. Si lo editamos les pagamos un tanto por ciento por cada copia vendida.

El sistema de colaboraciones ha progresado y ya son varios los programas y los programadores incorporados al sello **Dinamic**, **West Bank**, **Sgrizan** o **Phantomas** han sido producidos de esta forma.

«Estamos deseando que nos manden programas originales —nos decían. Queremos animar a todos los programadores a que así lo hagan. Lo más que les puede pasar es que reci-



ban una carta animándoles a continuar hasta que adquieran el nivel necesario para que les firmemos un programa. Nuestro lema sería que alguien en alguna parte podría llegar a ser programador, y que más de uno se ha llevado una sorpresa...»

Cuando les preguntamos por las características personales que debería tener un buen programador provocamos una larga discusión, pues las opiniones eran diversas. Pero en resumen todos están de acuerdo en que más que una superinteligencia lo que hace



falta es una enorme curiosidad y una paciencia inagotable para repasar, una vez tras otra, el programa en busca de errores, soluciones y mejoras. En cuanto a los medios, estiman que no es necesario tener ningún equipo especial —ellos lo demostraron en su momento— y nos cuentan que el **Phantomas** ha sido desarrollado sirviéndose su autor de un simple *cassette*.

De todas formas piensan que si el resultado es bueno, si un juego llega a los primeros puestos en las listas de ventas, cualquier esfuerzo tiene su compensación. «En este negocio hay muchas posibilidades, nosotros hemos empezado hace dos años, y aquí estamos. Cualquier programador de alto nivel viviría estupendamente haciendo dos o tres programas al año». Sólo con eso...

A la adaptación de programas para otras máquinas (**Commodore**, **Amstrand**, etc.) las consideran más fáciles que al desarrollo de programas originales, aunque son mucho más rutinarias, y aprovechan la ocasión para decirnos que «si hay programadores dispuestos a analizar versiones para otras máquinas, que nos digan su precio. A estos no les pagamos por copia sino que les compramos la versión con todos los derechos de explotación».

MICRODIGITAL S.A.

En sus nuevas oficinas, la mayor parte del espacio está reservada para los programadores. Allí se preparan nuevos programas con distintas versiones para otros ordenadores. De hecho, además de con el inevitable **PC** los **Sinclair** se codean allí con los **Amstrand** y los **Atari**. La presencia de estas máquinas nos hace pensar en el futuro.

- ¿Cuáles son vuestros planes? —les preguntamos.

- En abril de este año hemos constituido entre seis socios **Microdigital S.A.**, con el propósito de ofrecer una nueva imagen con vista al futuro, y una serie de acuerdos comerciales en distintos paises europeos».

De hecho, bajo la firma **Dinamic**, hace más de un año que debutaron en el mercado internacional con el **Saimazon** y **Babaliba**, aunque no debieron ser buenos comienzos. «Los ingleses nos sacudieron según llegamos» —y se rien, pero nos comentan que entonces no les hizo tanta gracia. «El caso es que con aquello aprendimos. Y es que los acuerdos con otros paises son muy difíciles. En realidad, se puede contratar lo que sea, pero a la hora de cobrar los *royalties* te encuentras con que no hay formas ni físicas ni legales de establecer controles. Así que no te queda otro remedio que fiarte».

Después del primer batacazo volvieron a insistir y, poco después, colocaron al **Rocky** en el número 2 de las listas inglesas. Actualmente están en tratos con diversas empresas en Francia, Alemania, Italia y Paises Bajos, y esperan firmar un contrato con una casa norteamericana.

A parte de estos ambiciosos proyectos y en lo que respecta a nuestro mercado, están desarrollando dos nuevas líneas de juego. De un lado, lo que ellos denominan «megagames» que son una serie de programas en dos cintas y de los que comentan «tendrán efectos termonucleares». Por cierto, que uno de ellos basado en el conocido *comic* de **Azpiri**, **Lorna y su robot**, puede dar mucho que hablar...

El otro proyecto, del que ya están en el mercado algunos ejemplos es una nueva línea fabricada bajo la marca **Future Stars**. «Se trata de un nuevo sello para editar los juegos de programadores nobeles que pueden presentar así su primer trabajo, aunque su nivel no sea tan alto como el de un programa **Dinamic**, y a unos precios competitivos».

EL MERCADO DEL SOFTWARE

Para terminar, antes de despedirnos, quisimos saber cómo se ve el tema de la piratería desde su nueva empresa. Nada más plantearles la cuestión se produjo otro controvertido debate entre ellos, y es que este asunto tiene varios puntos de enfoque. No obstante, como además de vender los programas también los crean, nos comentaron que «muchas veces cuando terminas un programa estás hecho migas por el esfuerzo y sin embargo sabes que un día después ya te lo habrán copiado y lo estarán vendiendo por ahí sin que tú veas ni una peseta. Es como trabajar a sueldo y que a fin de mes venga alguien y se lleve un buen tanto por ciento de tu nómina, así como así. A esta gente habría que recordarle que están deshaciendo el trabajo y el esfuerzo de los demás». En cuanto a la cuestión del precio aseguran que es el qué es precisamente por culpa de la piratería. Sin embargo no consideran que bajar los precios en estos momentos sea un buen método para hacerla disminuir, pues por un lado los piratas siempre pueden vender más barato, y por otro —y como ellos mismos pudieron constatar al sacar dos programas a la vez pero a distinto precio— «en este mercado se vende mejor lo más caro porque lo barato es, al parecer, sospechoso de falta de calidad».

Después de haber charlado largo rato con ellos tenemos la certeza de que la confianza en el futuro es una de las características de estos jóvenes empresarios. Asomados a la terraza de su despacho, con la vista impresionante sobre Madrid que ofrece un piso ventinueve, nos habían comentado parafraseando a alguien «desde aquí se tiene razón». Y es que los chicos de **Dinamic** no pueden ocultar su orgullo por llegar a donde han llegado, y no nos estamos refiriendo a sus oficinas en el piso ventinueve.

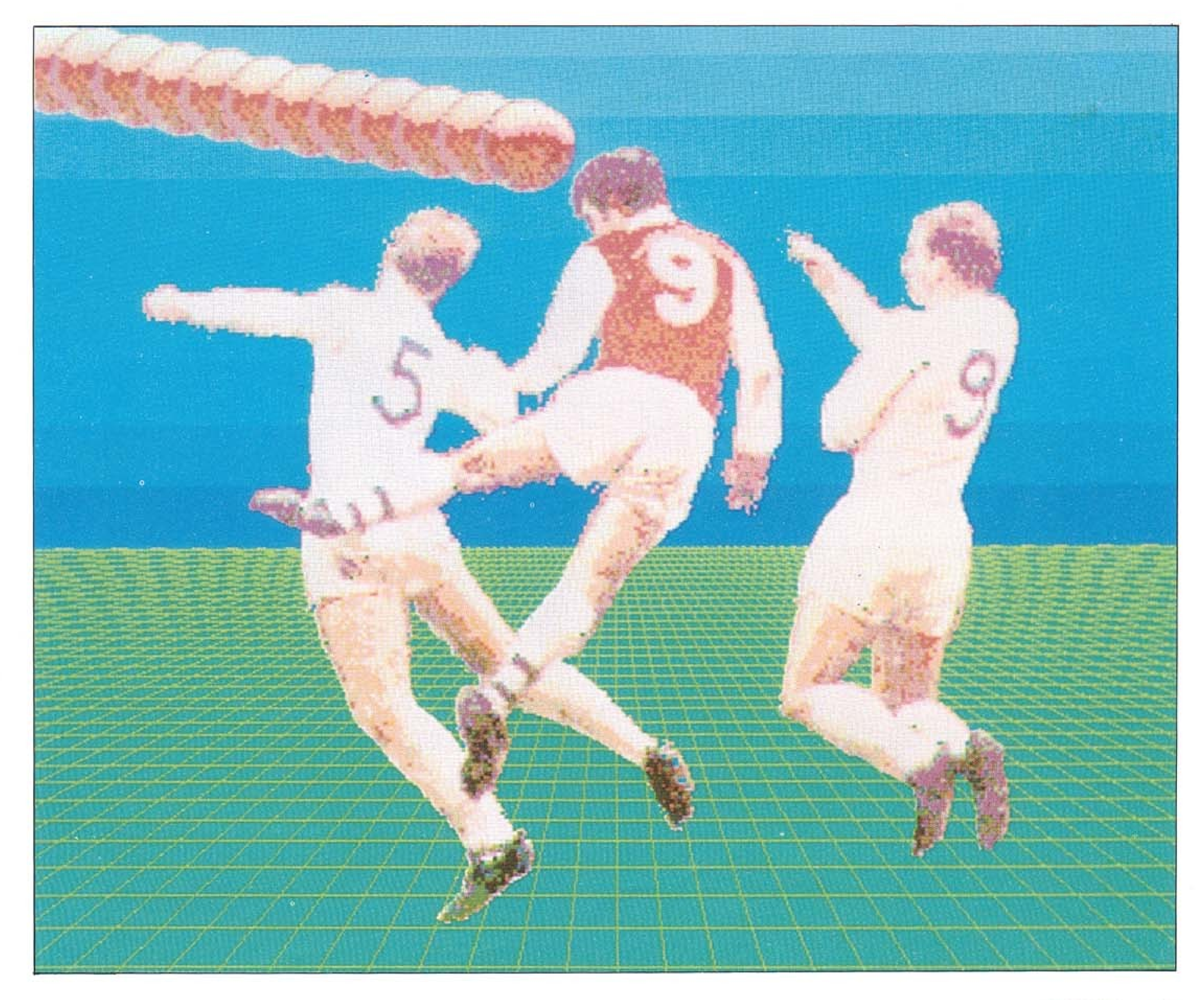
PREDICIENDO LO IMPREDECIBLE

- LAS REGLAS DEL JUEGO
- SACAR LOS NUMEROS DE UNA CAJA O UN SOMBRERO
- NUMEROS ALEATORIOS
- MUESTRAS Y MUESTREO

¿Qué posibilidades tienes de acertar un pleno en las quinielas de la próxima temporada? Puede que no mejores tu suerte, pero al menos entenderás porqué pierdes.

La instrucción de un novato en el manejo de un moderno avión comercial o de un bombardero sería enormemente cara al tener en cuenta factores tales como el combustible, las pistas de aterrizaje y toda la infraestructura de apoyo. Por eso las autoridades militares opinan que es más barato gastar dinero en aviones de entrenamiento y simuladores, enormes artefactos controlados por ordenador que nunca abandonan el suelo, pero que dan una impresión de vuelo de lo más realista.

Este mismo principio resulta cierto en un gran número de circunstancias, incluída la programación de juegos, el desarrollo industrial y el *marketing*, así como la investigación científica y los estudios gubernamentales. Es preferible programar un ordenador para predecir, por ejemplo, el resultado probable de un fallo estructural o una determinada política económica, que realizar el experimento real que podría requerir cinco años o incluso más en determinadas circunstancias. No



¿BUSCAS LOS MEJORES VIDEOJUEGOS?



te los ofrece MENSUALMENTE

El mejor soft para los lectores de 

PARA USUARIOS DE SINCLAIR SPECTRUM 48 K

OFERTA ESPECIAL

Nº 004
P.V.P. 995 Ptas.

Nº 005
P.V.P. 995 Ptas.

Nº 006
P.V.P. 995 Ptas.

KRYPTON RAIDERS

Sus pies resbalan en la dura roca, un sudor frío le cruzaba la cara, frente a él se hallaba la gruta del planeta "KRYPTON". El sabía lo que esto significaba, internarse en un mundo hostil donde el arma láser no debía dejar de actuar. Pero su mente estaba serena, y recordaba perfectamente la misión, recoger los cuatro diamantes símbolo de su mundo y alcanzar el helicóptero para huir lo antes posible.

NO LO DUDES, LANZATE A LA AVENTURA.

ALI - BEBE

El pequeño ALI - BEBE, se halla perdido en un laberinto plagado de enemigos en donde fue encerrado por su malvado padrastro.

Para salvarse, debe encontrar la lámpara mágica en la que se encuentra su genio protector.

ALI - BEBE dispone únicamente de 5 vidas y 15 biberones, los cuales puede arrojar contra sus feroces enemigos, de los que se librará sólo por unos instantes.

Si te sientes con fuerza, adelante, y ¡BUENA SUERTE!.

TOMMY

El juego consiste en recoger las notas que hay en cada una de las aulas del instituto: MATEMATICAS, CIENCIAS, QUIMICA, LABORATORIO, LITERATURA y depositarlas en el aula de VIAJE DE FIN DE CURSO.

¡PIDE ESTAS 3 NOVEDADES Y PAGA SOLO 2!

¿Tienes un SPECTRUM 48 K?

PIDE LAS 3 NOVEDADES DEL MES Y PAGA SOLO 2 ¡Ahórrate 995 ptas.!

Para todo el que posea uno de estos magníficos ordenadores personales, la oportunidad de estar más al día ha llegado, por fin. **CLUB DE ELITE**, el mejor soft para los lectores de **INPUT**, te ofrece a partir de ahora lo mejor y más actual del software mundial, siempre con excepcionales ofertas.

¡Todo son ventajas en este CLUB!

- *No hay cuota alguna de entrada. Basta ser lector habitual de INPUT SINCLAIR.*
- *No hay obligación de compras mínimas por año.*
- *Podrás comprar soft (programas y/o juegos individualmente), siempre con un **10% de DESCUENTO**, o acogerte a la oferta del mes de 3 JUEGOS AL PRECIO DE 2.*

UN GRAN CLUB PARA TI, AMIGO LECTOR

Lo único que has de hacer es comprar INPUT SINCLAIR cada mes y ver las ofertas que en este CLUB DE ELITE se ofrecerán. Las novedades de cada mes enriquecerán el fondo del CLUB, pues seguirán ofertándose, ya individualmente, en los meses sucesivos.

ENVIA EL CUPON ADJUNTO HOY MISMO

Te enviaremos los juegos por ti escogidos, ya sea la oferta del mes o cualquiera de ellos individualmente, **directamente a tu domicilio,** sin cargo adicional alguno.



ENVIA ESTE CUPON DEBIDAMENTE FRANQUEADO A:
EDISA, López de Hoyos, 141
28002 Madrid,
o bien llámanos por teléfono al (91) 415 9712

CUPON DE PEDIDO

SI, envíenme contra reembolso los juegos de la oferta del mes, cuyos números les indico en las casillas correspondientes, por los que me facturarán el precio de 2 juegos solamente.

[ ] [ ] [ ] GRATIS

SI, envíenme únicamente y contra reembolso el juego cuyo número indico en la casilla correspondiente, con un descuento del 10% sobre el P.V.P.

[ ]

NOMBRE
DOMICILIO
NUM. PISO ESC. COD. POSTAL
POBLACION PROV
POR FAVOR, FIRMA AQUI

S 7-86

hay que sorprenderse pues de que la simulación sea una actividad que resulta especialmente impulsada. Pero debido a su complejidad sólo ha podido ser posible con la llegada de los ordenadores. Actualmente, hasta los micros domésticos pueden realizar un cierto grado de simulación.

Una de las principales herramientas de este tipo de creación de modelos por ordenador es el conjunto de reglas que nos proporciona el estudio matemático de la probabilidad.

El resto del modelo se deriva de un análisis de la información estadística, obtenido a partir de una descripción de la situación real.

Con unas cuantas reglas sencillas, el usuario del microordenador doméstico puede predecir las alternativas futuras para un sinfín de sucesos diferentes, si bien la fiabilidad de los resultados depende de la precisión de los datos introducidos en el programa o de la precisión de las reglas por las que se rige el juego. Se puede demostrar este hecho mediante la simulación de los resultados de los partidos de fútbol.

Todos los fines de semana de la temporada futbolística, millones de aficionados a las quinielas están pendientes de los resultados a la espera de conseguir algún premio sustancioso. Por desgracia, todos menos unos pocos se ven decepcionados y tienen que esperar una semana más para tener otra oportunidad. Si no quieres tener que esperar una semana entera para tener otra oportunidad, teclea el siguiente programa y prueba tu suerte todas las veces que quieras:

```
10 POKE 23658,0
20 DIM A$(14,10): DIM B$(14,
   10): DIM R$(3,9)
30 BORDER 0: PAPER 0: INK 7:
   CLS
50 PRINT AT 0,5; INVERSE 1;
   " APUESTA FUTBOLISTICA "
100 PRINT : PRINT AT 5,5;
    "ESCRIBE LOS ENCUENTROS"
    : PRINT
110 FOR D=1 TO 100: NEXT D
120 FOR K=1 TO 14
140 INPUT "EQUIPO LOCAL ";A$
    (K): PRINT TAB 8;A$(K);
150 INPUT "EQUIPO VISITANTE
    ";B$(K): PRINT TAB 18;
    "-";B$(K)
160 NEXT K
170 CLS : PRINT  FLASH 1;
    PAPER 2;AT 10,6;
    "PARTIDOS DISPUTANDOSE"
180 FOR V=1 TO 1000: NEXT V
190 CLS : PRINT AT 0,8;
    INVERSE 1;"[2*
    ESPACIO] RESULTADO
    [2*ESPACIO]"
200 LET R$(1)="LOCAL":
    LET R$(2)=
    "VISITANTE"
210 LET R$(3)="EMPATE"
220 PRINT : PRINT :
    PRINT "[5*ESPACIO]
    ENCUENTRO";
    TAB 22;"VICTORIA":
    PRINT
225 FOR I=1 TO 14
230 LET Y=RND*1
240 IF Y<=.5 THEN LET Z$=R$
    (1)
250 IF Y>.5 AND Y<=.75 THEN
    LET Z$=R$(2)
260 IF Y>.75 AND Y<=.9 THEN
    LET Z$=R$(3)
280 PRINT A$(I);"-";B$(I)
    ;Z$
290 NEXT I
300 PRINT : PRINT
310 INPUT "OTRA VEZ? (S/N)";
    T$
320 IF T$="N" THEN GO TO 340
330 GO TO 30
340 STOP
```

Aunque en esta simulación futbolística no hay reparto de dividendos, puede que te comunique una sensación bastante realista. En cualquier caso servirá para demostrar lo pequeña que es la posibilidad de conseguir 24 puntos.

Ejecuta el programa.

El mal tiempo obliga a veces a suspender determinados partidos. Cuando esto sucede, un grupo de expertos futbolísticos decide qué resultados se habrían producido de haberse jugado realmente el encuentro. De hecho, tanto el programa como el grupo de



expertos son simuladores; proporcionan una representación simbólica de un proceso real.

El programa no es especialmente difícil. Una vez introducidos los encuentros, la parte principal de la simulación decide el resultado de cada partido y lo presenta en pantalla.

DECIDIENDO LOS RESULTADOS

La regla en que se basa el ordenador para decidir cualquier resultado, es la vieja regla de las quinielas de que la mitad de los resultados son victorias domésticas, las victorias fuera de casa y los empates son el resto. Esta pro-



porción puede variar en las líneas 240, 250 y 260.

GENERACION DE NUMEROS ALEATORIOS

Hace años se utilizaban métodos manuales o mecánicos, tales como lanzar dados, barajar cartas o emplear una ruleta, para generar listas de números aleatorios. Estos procedimientos eran lentos y tediosos, por lo que el famoso matemático norteamericano **John Von Neuman** propuso la técnica del cuadrado medio. Se empieza con un número de cuatro cifras (la semilla), el siguiente número aleatorio se obtiene multiplicando la semilla por sí misma y quedándose con los cuatro dígitos centrales. Por ejemplo, supongamos que la semilla es 5272. El segundo número se genera tomando los cuatro dígitos centrales de (5272 ↑ 2) que es 27 793 984. El resultado (7939) es prácticamente aleatorio. Se puede obtener un segundo número elevando al cuadrado el número 7939, continuando así el proceso.

Naturalmente, surge la pregunta: ¿Cómo puede un proceso matemático, que es esencialmente repetitivo, producir números verdaderamente aleatorios? La respuesta es que en realidad no puede hacerlo. Sin embargo los números así obtenidos se comportan como si realmente fueran aleatorios, por lo que generalmente se les llama pseudoaleatorios o quasi-aleatorios. De hecho los números generados utilizando la función RND son pseudoaleatorios. Lástima que esta técnica del cuadrado central no sea muy útil para la generación de números aleatorios por ordenador.

Aparte de ser lenta, la secuencia se repite en seguida y en cuanto se ob tiene un cero se repite todo el proceso. La mayoría de los micros domésticos utilizan un método de **congruencias**, que utiliza los restos para generar secuencias pseudoaleatorias. El siguiente programa utiliza una sencilla fórmula y la función INT (parte entera) para proporcionar un ejemplo de esta técnica:

```
20 BORDER 0: INK 7: PAPER 0:
   CLS
30 PRINT AT 0,4; INVERSE 1;
   " NUMEROS PSEUDO
   ALEATORIOS "''
40 INPUT " CUANTOS NUMEROS?"
   ;N: LET S=0
50 LET X=.677829*PEEK 23673/
   50
60 LET X=X*1842.95
70 LET X=X-INT (X)
80 LET P=INT (X*1000): PRINT
   "[5*ESPACIO]";.001*P,
90 LET S=S+1
110 IF S<=N THEN  GO TO 60
120 STOP
```

La línea 50 utiliza la función de hora del ordenador para establecer como punto de partida un número diferente en cada ejecución. El número .677829 es una constante arbitraria. Después de multiplicar el valor de partida por otra constante (línea 60), se obtiene el resto de la parte decimal. Cambia el valor de las constantes de las líneas 50 y 60 y ejecuta nuevamente el programa para ver qué tipo de resultados obtienes.

En muchos casos es preferible —y bastante más sencillo— utilizar la función RND que lleva incorporada tu ordenador. Al modificar el valor de x en la expresión RND(x) podrás seleccionar una secuencia que se repita, lo cual es una buena prueba para reinicializar la «semilla» cada vez.

Quizá el punto más importante que tienes que recordar cuando escribas programas es que la función RND lo que te da es una variable aleatoria y no un valor algebraico

MUESTRAS Y OBSERVACIONES

Los sondeos preelectorales, las organizaciones de consumidores y los gobiernos utilizan los ordenadores para generar muestras aleatorias. Por ejemplo, es importante que una empresa dedicada a la investigación de mercados y entrevista a 1000 personas, obtenga una muestra típicamente variada, que pueda representar a un grupo mucho mayor de población. No sería válido por ejemplo elegir a todos los individuos entre los miembros de un club de coches antiguos, cuando lo que se pretende investigar son las costumbres nacionales en materia de preferencias automovilísticas.

La mejor forma de hacer que una muestra sea representativa es obtenerla aleatoriamente, con lo que se excluye todo tipo de sesgo o tendencia.

Esto sigue siendo cierto también cuando se trabaja con una muestra más pequeña, como podría ser el caso de un club de amigos de los coches an-

tiguos o de los lectores de **INPUT**. También en este caso es importante la selección aleatoria, por ejemplo en el caso en que quisieras determinar las preferencias de los lectores de las revistas de informática.

También puede ser que quieras extraer una muestra de tu propia lista de miembros de un club informático. El proceso de muestreo es básicamente similar a la simulación de una extracción de papeletas de un sombrero. No obstante en la simulación, cada papeleta vuelve a ser introducida en el sombrero antes de proceder a la siguiente extracción. En el muestreo, cada papeleta seleccionada se deja fuera del sombrero y se realiza la siguiente extracción. Teclea el siguiente programa para ver cómo se utiliza RND para generar una muestra aleatoria:




```
10 DIM B$(10,16)
20 DIM A$(10,16)
30 BORDER 0: PAPER 0: INK 7:
   CLS
50 PRINT AT 0,7; INVERSE 1;
   " MUESTREO ALEATORIO "
90 PAUSE 100: CLS
100 FOR I=1 TO 10: READ A$
    (I): NEXT I
110 INPUT " MEDIDA DE LA
    MUESTRA? ";N
120 FOR V=1 TO 10: LET B$(V)
    =A$(V): NEXT V
130 FOR J=1 TO N
140 LET R=1+INT (RND*10)
150 IF B$(R)="[16*ESPACIO]"
    THEN  GO TO 140
160 PRINT B$(R)
170 LET B$(R)=""
180 NEXT J
190 INPUT " OTRA MUESTRA
    (S/N)? ";G$
200 IF G$="S" THEN  CLS :
    GO TO 110
210 STOP
220 DATA "BONN","COPENAGUE",
    "LONDRES"
230 DATA "MADRID","MOSCU",
    "NUEVA YORK"
240 DATA "PARIS","ROMA",
    "ESTOCOLMO","VIENA"
```

Después de leer los datos (línea 100) se genera un número entero aleatorio R comprendido entre 1 y 10 (línea 140). A continuación tu ordenador imprimirá el elemento R-ésimo de la lista (línea 140). Una vez que ha sido seleccionado un artículo, hay que eliminarlo del banco de datos para que no resulte elegido por segunda vez. La línea 170 se ocupa de esto. Observa que podrías utilizar RND(10) para generar una variable aleatoria dentro del margen de 1 a 10, si bien el método de la línea 140 se deriva del uso de las fracciones decimales para probar el margen de probabilidades.

Normalmente, el colectivo de datos del que se obtiene la muestra es bastante más extenso que los diez elementos de los que consta este ejemplo. Para muestrear semejantes volúmenes de datos, el programa anterior sería lento e ineficiente. Supongamos por ejemplo que un encuestador de-

MI COMPUTER 1

CURSO PRACTICO DEL ORDENADOR PERSONAL, EL MICRO Y EL MINIORDENADOR



SUPER OFERTA DE LANZAMIENTO

Con su primer fascículo recibirá GRATIS el fascículo n.º 2, es decir, su primer envío constará de 5 fascículos al precio de 4.

RECORTE ESTE CUPON Y ENVIELO A EDISA (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid

SI, deseo suscribirme a **MI COMPUTER** y recibiré en mi hogar 4 fascículos al mes, abonando sólo **700 Ptas.** por cada envío. El servicio a mi domicilio es GRATUITO.

▽ POR FAVOR, RELLENE SUS DATOS EN MAYUSCULAS ▽

NOMBRE ______

APELLIDOS ______

DIRECCION ______ N.º ______

PISO ______ COD. POSTAL ______

CIUDAD ______

PROVINCIA ______

TLFNO. ______

FIRMA ______

Editorial Delta, S.A.

sea extraer 200 nombres de un censo electoral de 60000 votantes de una determinada circunscripción electoral; un programa como el anterior necesitaría buscar en toda la lista de electores 200 veces. Para evitar la larga espera que esto llevaría aparejado, sería mejor utilizar un método de un solo paso.

LA BUSQUEDA DE UN SOLO PASO

En el método de un solo paso se leería toda la lista electoral de arriba a abajo una sola vez. Al ir considerando cada nombre, se toma la decisión de si incluir o no a esa persona en la muestra. Este método resulta fácil de programar; realiza los siguientes cambios en el último programa y vuelve a ejecutarlo:

```
50 PRINT AT 0,1; INVERSE 1;
   " MUESTREO DE UN SOLO
   PASO "
120 LET A=N: LET C=10
130 FOR J=1 TO 10
140 IF A=0 THEN  GO TO 190
150 IF RND*1<=A/C THEN PRINT
    A$(J): GO TO 170
160 LET C=C-1: GO TO 180
170 LET A=A-1: LET C=C-1
```

Si ahora tecleas un 3, para seleccionar tres elementos de la lista, el primero de ellos (Bonn) es el primero que se considera. Si la función RND (línea 150) es menor que 3/10, será seleccionado Bonn. Copenague es el seleccionado a continuación. Si Bonn ya está en la selección, Copenague sólo podrá ser seleccionado si la función RND genera un valor menor que 2/9. Por otra parte, si todavía no ha sido seleccionado Bonn, las probabilidades de Copenague se elevarán hasta 3/9. Las líneas 160 y 170 actualizan las probabilidades. Cuando compares los resultados de las selecciones de este programa con las del anterior, advertirás que el método de un solo paso da las muestras en orden alfabético.

A primera vista podría parecer que con una lista de sólo diez ciudades, el número de posibles muestras es pequeño. Esto no es cierto. De hecho hay 120 posibles muestras diferentes de tamaño 3 y 252 muestras de tamaño 5. En un próximo artículo veremos de qué forma pueden desarrollarse estas ideas para ser utilizadas en otros tipos de simulación con diversas aplicaciones.

NO OLVIDES EL TELEFONO... 

Cuando, por cualquier motivo, nos escribas, no olvides indicar tu número de teléfono. Así nos será más fácil y rápido ponernos en contacto contigo. Gracias.

Tú sabes lo que es "ser un LIDER"

VESPINO ALX EL LIDER



*Para ser el Mejor hay que hacer bien las cosas.
En su momento.
La Alta Tecnología de MOTOVESPA ha perfeccionado su VESPINO, ampliando sus prestaciones, para poder seguir siendo el Mejor. Vive hoy con tu VESPINO.
Con tus amigos, tu gente. Disfrutando. En tu quehacer diario, en el deporte.
Tú que sabes ganar, gana con VESPINO.
Tú sabes lo que es "ser un LIDER".*

VESPINO

VESPA ha elegido





4.000 puntos de venta y asistencia técnica

HACER SALTAR A LA BANCA

- EL ORDENADOR Y EL JUEGO DE CARTAS
- JUGANDO CON EL ORDENADOR
- COMO FUNCIONA EL PROGRAMA
- LAS PIEDRAS

Para completar el juego de las veintiuna, tienes que programar tu ordenador para que juegue cuando le toque el turno. Aquí tienes además una descripción de las reglas del juego para el caso de que no estés familiarizado con el mismo.

Antes de pasar a examinar el resto de la programación, vale la pena recapitular las reglas del juego.

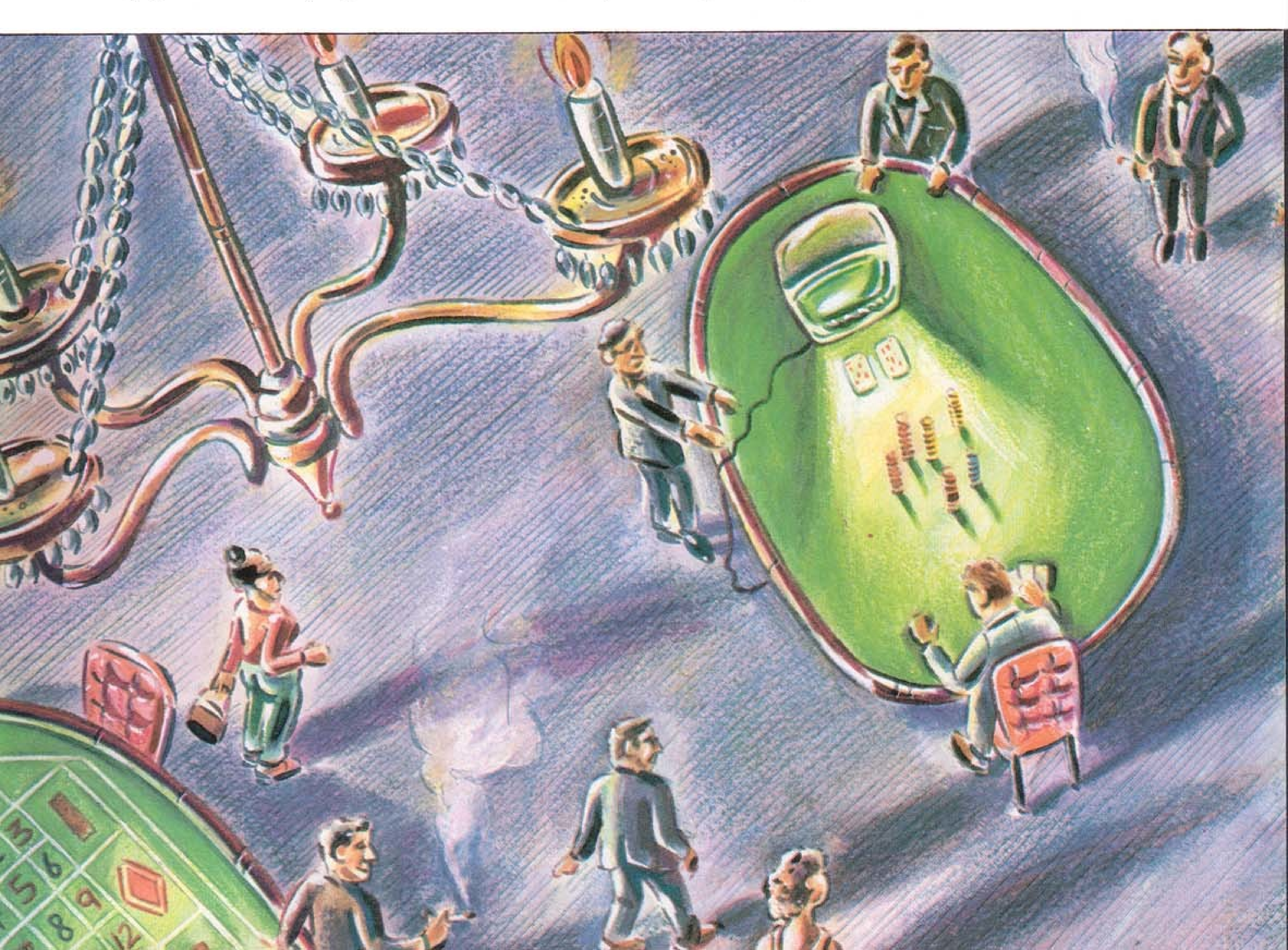
El juego de las veintiuna se juega con un mazo ordinario de 52 cartas. Las cartas del 2 al 10 valen lo que indica su número, las figuras valen 10 y el as vale 1 u 11, según las necesidades del jugador. En este juego el ordenador se ocupa de ir llevando la cuenta de los totales, por lo que no tienes que hacerlo tú.

Normalmente se juega con dinero, o con «piedras», pero en esta versión para ordenador hay que programarle para que cuente la puntuación en fichas imaginarias; con las otras fichas un programador poco escrupuloso podría hacer que la máquina dejara de ser honrada. El ordenador se programa para actuar como banquero en todo momento y será siempre el encargado de repartir las cartas.

Al principio del juego se barajan las cartas y se sacan dos de ellas, poniéndolas boca abajo. En la pantalla aparecerá boca arriba la carta del jugador, aunque el programa ha sido diseñado para que la máquina no sepa qué carta es la que tiene el jugador.

El jugador debe apostar ahora sobre esta primera carta, antes de que se saque una nueva para cada uno.

El objeto del juego es terminar con mejor puntuación que la banca, es decir con un valor total más elevado. Una mano que sume en total más de 21, se «pasa» y pierde. Una mano con puntuación entre 16 y 21 solamente vence a la banca si la máquina tiene una mano más baja o se ha pasado de 21. Sólo hay dos triunfos especiales: las veintiuna, consistentes en un as y



un diez o una figura, que en total suman 21 con sólo dos cartas, y las cinco cartas, que es cualquier grupo de cinco cartas con un total de 21 o menos. Las veintiuna del jugador vencen a todo lo que tenga la banca excepto a otras veintiuna. Las cinco cartas también vencen a todo lo que tenga la banca excepto a las veintiuna y a otras cinco cartas.

FUNCIONAMIENTO DEL PROGRAMA

Después de que el jugador ha recibido una segunda carta, el programa comprueba si se han hecho veintiuna. De no ser así, al jugador se le ofrecen tres opciones. En primer lugar, si totaliza una puntuación de 16 o mayor, y está satisfecho con sus cartas, puede decidir «plantarse». Aquí termina su turno y el juego continúa por parte de la banca. En segundo lugar, si el total está por debajo de 16 o el jugador no está satisfecho con el total, puede adquirir más cartas «robando» o «doblando». En el juego normal de las veintiuna pedir una carta impone al jugador la obligación de hacer una puesta igual a la original. En los cambios el banquero utiliza la carta que hay boca abajo. En la versión para ordenador no se hace distinción entre las cartas boca arriba y boca abajo, por lo que cada carta aparece al lado de la anterior. El robo no implica que el jugador tenga que hacer una nueva puesta, y en un juego real la carta se colocaría boca arriba; aquí no hay diferencia. El jugador no puede doblar después de robar y si está intentando conseguir la última carta para completar un triunfo de cinco cartas, tiene que robar obligatoriamente. Después de extraer cada carta el programa comprueba que ni el jugador ni la banca han superado los 21 puntos y por tanto no se han pasado.

Si el jugador se pasa, la banca toma toda su puesta y el ordenador ya no tiene que ocuparse de su juego. En caso contrario el banquero da la vuelta a las cartas y tiene que decidir si plantarse o tomar más cartas. Si logra las veintiuna en esta fase, gana automáticamente y ya no se sacan más cartas. Si no tiene las veintiuna, la banca seguirá sacando nuevas cartas hasta que quede satisfecha con el total, consiga un triunfo de cinco cartas o se salga.



Si la banca se planta con un numero inferior a 21 y no ha logrado un triunfo de cinco cartas, aparece en la pantalla un mensaje y un número igual al tanteo de las cartas actuales más uno. Si el tanteo es 21, aparece el mensaje SOLO VEINTIUNA O LAS CINCO CARTAS. Si la banca consigue las cinco cartas, el jugador recibe el mensaje SOLO LAS VEINTIUNA. En un juego ordinario, los jugadores tendrían que declarar al banco sus cartas en caso de que ganaran, pero en este juego, naturalmente, el ordenador ya tiene la puntuación del jugador.

Si el jugador consigue reunir una puntuación igual a la anunciada, gana, pero si tiene menos pierde su puesta. Cuando el jugador gana, el banco pone una cantidad igual a la apostada por el jugador, por lo que éste recibe el doble de la cantidad que se jugó. A diferencia de lo que ocurre con las reglas de otros juegos, las veintiuna y las cinco cartas no se pagan con más fichas.



Otra diferencia con respecto a los juegos ordinarios de cartas, es que el reparto no puede pasar al jugador, debiendo ser siempre el ordenador el que «da» las cartas. En el juego ordinario el banco cambia cuando se logran las veintiuna, en este juego el banco es siempre el ordenador, por lo que siempre tiene ventaja sobre el jugador; en las veintiuna ordinarias, la banca tiene también ventaja sobre el jugador. Por ello el jugador tiene ante sí la difícil tarea de hacer saltar a la banca para ganar, lo que aquí significa acumular más de 1000 piedras. El jugador empieza a jugar con 100 piedras y no debe perderlas todas, ya que de ocurrir así perdería el juego completo.

Carga ahora las dos últimas partes del programa y teclea las siguientes líneas para tener completo el juego de las veintiuna:

Teclea

```
2500 PAUSE 50
2510 LET DPF=B: LET AF=B:
     LET X=C: LET Y=10
2520 FOR J=C TO 2
2530 LET Z=O(J): GO SUB 5500
2540 IF VA>10 THEN LET VA=10
2545 FOR K=C TO 2: LET W(K)=
     W(K)+VA: NEXT K
2546 IF VA=C AND AF=B THEN
     LET W(2)=W(2)+10: LET
     AF=C
2550 LET X=X+6: NEXT J
2560 IF W(2)=21 THEN PRINT
     PAPER 2; INK 7; AT 14,
     18;" VEINTIUNO! "; AT
     15,17;" GANA LA BANCA "
     : LET DPF=C: GO TO 2000
2600 IF PF=C THEN GO TO 2720
2610 PAUSE 75
2630 IF W(C)>21 THEN PRINT
     PAPER C; INK 7; AT 21,B
     ;" LA BANCA PIERDE! ";:
     GO TO 2740
2635 IF W(2)>21 THEN LET W
     (2)=W(C)
2640 IF W(2)<16 THEN GO TO
     2800
2650 IF FF=C THEN GO TO 2800
2660 LET PR=(W(2)-8)/13
2670 IF PR<RND THEN GO TO
     2800
2700 IF W(2)>21 THEN LET W
     (2)=W(C)
2710 IF W(2)=21 THEN PRINT
     AT 20,B;" LA BANCA PAGA
     SOLO VEINTIUNA O EL
     JUEGO DE CINCO CARTAS":
     GO TO 2000
2720 PRINT  PAPER 2; AT 21,B
     ;" LA BANCA JUEGA ";W
     (2)+C;
2725 IF S(2)>21 THEN LET S
     (2)=S(C)
2730 IF W(2)>=S(2) THEN PRINT
     PAPER 2;" Y GANA ": GO
     TO 2000
2740 PRINT PAPER 2;" -TU
     GANAS! ": LET CP=CP+BET
     *2: GO TO 2000
2800 LET Z=C(CC): GO SUB 5500
2805 IF VA>10 THEN LET VA=10
2810 FOR K=C TO 2: LET W(K)=
     W(K)+VA: NEXT K
2820 IF VA=C AND AF=B THEN
     LET W(2)=W(2)+10: LET
     AF=C
2822 IF W(1)<22 AND X=25 THEN
     PRINT PAPER 2; INK 7; AT
     21,B;" CINCO CARTAS! ":
     GO TO 2000
2825 GO SUB 7000
2830 LET X=X+6: GO TO 2610
```

Después de una pausa de un segundo en la línea 2500, empieza el programa con el turno del jugador. La línea 2510 inicializa el indicador de veintiuna de la banca, pone a cero el indicador de ases y asigna valores a las coordenadas de la esquina de la primera carta de la banca.

DANDO LAS CARTAS

Las dos cartas previamente almacenadas se reparten en el bucle FOR ... NEXT que se extiende entre las líneas 2520 y 2550. La línea 2530 representa

la carta. La línea 2540 asigna a las cartas con figura el valor 10 antes de que la línea 2545 sume el valor de la carta sacada al total; caso de que haya presente un as, habrá dos totales. Si una de las cartas es un as, en la línea 2546 se suma 10 a W(2) y se activa el indicador de as. La posición de la siguiente carta se ajusta en la línea 2550.

La línea 2560 comprueba si se han hecho veintiuna y anuncia que gana la banca si así es. Se activa el indicador de veintiuna de la banca. Si el jugador ha hecho veintiuna —línea 2600— el programa salta a la rutina de control de resultados: qué tiene cada uno y quién gana.

Después de una pausa en la línea 2610, en la línea 2630 se comprueba si la banca tiene más de 21 puntos; cuando esto ocurre se envía el mensaje: LA BANCA PIERDE. La línea 2635 comprueba si el mayor de los dos totales —cuando se presenta un as— es mayor que 21. El programa se concentra entonces en el más bajo de los dos totales. Si la banca tiene menos de 16 puntos, se toma una nueva carta en la línea 2800.

¿OTRA CARTA MAS?

Las líneas 2660 y 2670 deciden si el jugador desea otra carta más o no. Esta decisión no se toma con arreglo a una norma determinística. No sería una buena idea imponer un límite estricto sobre cuándo tiene que plantarse la banca, ya que el jugador en seguida sabría exactamente lo que que tendría que hacer para vencer a la máquina. En lugar de esto, hay que introducir cierto factor de aleatoriedad para que sea imposible predecir con exactitud con qué número se plantará la máquina.

El problema al intentar escribir este tipo de rutinas es conseguir que el ordenador actúe un poco como un ser humano. Piensa cómo actuarías tú en el juego: será bastante menos probable que pidas otra carta cuando tienes 20 que cuando tienes 16, ya que las probabilidades de pasarte son mayores. Al comparar PR con un número aleatorio hay un factor de aleatoriedad y una corrección ponderada con arreglo a la puntuación que tiene el banco.

PUNTUACION TOTAL

Si W(2) es superior a 21, la línea 2700 permuta W(2) y W(1). Si la banca tiene 21, la línea 2710 presenta el mensaje LA BANCA PAGA SOLO LAS VEINTIUNA O EL JUEGO DE CINCO CARTAS.

En la línea 2730 se compara la puntuación de la banca con la del jugador. Si gana la banca, la línea 2730 presenta un mensaje, pero si es el jugador el que gana, el mensaje lo presenta la línea 2740. Cuando gana el jugador se suma el total en la línea 2740.

La sección final del programa, líneas 2800 a 2830, se ocupa de darle al jugador cartas tomadas del mazo.

La línea 2805 asigna el valor 10 a las figuras. En la línea 2810 se suma el valor de la carta a W(1) y W(2). Los ajustes para los ases se hacen en la línea 2820, antes de que la línea 2822 compruebe si se han hecho las cinco cartas. El puntero de cartas se ajusta llamando a la subrutina de la línea 7000 y la posición de la siguiente carta se calcula en la línea 2830.



CONVIERTE TU AVENTURA EN UNA EPOPEYA

- COMO SE ALMACENA EL TEXTO
- FORMAS DE AHORRAR ESPACIO
- LOS CODIGOS ASCII
- LISTADO DE ENSAMBLADOR

Si eres muy hábil con los juegos de aventuras, pero te sientes frustrado por las limitaciones de memoria de tu máquina, ¿qué te parecería disponer de un programa que permitiera acomodar la misma cantidad de texto en mucho menos espacio?

El mayor problema con el que te encontrarás al escribir programas de juegos de aventuras es que nunca tienes suficiente espacio de memoria para albergar tu última obra maestra. Aparte de comprarte una máquina nueva o una ampliación de memoria, es muy poco lo que puedes hacer, como no sea achicar los horizontes de tu mundo de aventura o simplificar el programa.

La única alternativa que tienes para disminuir la presión ejercida por la escasez de memoria es intentar encontrar la manera de que el programa ocupe menos espacio. Los métodos habituales para acortar programas, no te serán de mucha utilidad, ya que casi todo el programa es texto. Lo que necesitas es precisamente un método que te permita almacenar la mayor cantidad de texto en el menor espacio posible.

En los ordenadores el texto se almacena normalmente en forma de código ASCII. El uso del código ASCII hace que cada carácter ocupe ocho bits, es decir un byte de espacio de memoria.

Si se pudiera apretar cada carácter de forma que ocupe menos de ocho bits, se ahorraría espacio de memoria. Para conseguirlo dispones de varios métodos, cada uno con sus ventajas y sus inconvenientes.

Probablemente la manera más fácil de comprimir el texto es utilizar sólo una parte de la codificación ASCII. Si te conformas con un reducido número de caracteres, por ejemplo sólo mayúsculas, números y algún signo de puntuación, resulta muy fácil almacenar cada carácter en seis bits.

Por ejemplo, si eliges el conjunto de caracteres ASCII que va desde 20 a 5F en hexadecimal, tendrás un buen juego de caracteres para utilizar dentro de tu programa. Si restas 20 hexadecimal del valor ASCII de cada uno de los códigos, el rango quedará reducido a 0 - 3F hexadecimal, que puede almacenarse con sólo seis bits.

El uso de este tipo de codificación te permitirá reducir las exigencias de memoria en una cuarta parte, lo que significa que podrás colocar cuatro caracteres en el mismo espacio en que anteriormente colocabas tres. Para decodificar los caracteres almacenados, no tienes más que ir tomando los bits de seis en seis y sumar 20 hexadecimal al número resultante para obtener el valor ASCII del carácter original.

El mayor inconveniente de una codificación de este tipo es que te ves obligado a elegir un margen de 40 caracteres (en hexadecimal) antes de empezar a programar, debiendo respetarlo estrictamente. Como te encuentres que te hace falta un carácter que no pertenezca al grupo seleccionado, lo vas a tener bastante difícil. Pudiera ocurrir que utilizar únicamente letras mayúsculas resulte inaceptable para los usuarios de máquinas que normalmente hacen aparecer en su pantalla letras minúsculas, por lo que deberás encontrar otra forma de disponer las cosas. En resumen, tienes que buscar otra alternativa.

Existe una posibilidad que se parece a la forma en que opera la lengua china, consistente en asignar un único carácter o valor numérico a cada palabra. Este tipo de codificación te impone la necesidad de decidir el vocabulario completo de tu aventura, asignando a cada palabra un número. La codificación consiste en comparar cada palabra con una lista de datos y almacenar cada código en la memoria. La decodificación es la operación inversa.

Con la «solución china» es posible lograr un uso muy eficiente de la memoria, pero tiene el inconveniente de que hay que volver a escribirla para cada nueva aventura. Naturalmente, siempre puedes empezar con un vocabulario muy extenso y albergar la esperanza de haber elegido las palabras suficientes para tus necesidades, pero el precio de esto es una gran cantidad de memoria. Podrías llegar a encontrarte utilizando números muy grandes como códigos, y la exploración de las listas resultaría muy lenta.

La mejor solución de todas sería algún sistema de compresión de texto que pudiera valer para cualquier aventura; de hecho tendría que valer para cualquier texto que pudieras encontrarte. Dicho «compresor» debería hacer un uso lo más eficiente posible de la memoria disponible, en función de la manera de comprimir el texto y de la cantidad de espacio disponible por el propio *software* de compresión.

Es muy importante que el compresor de texto sea capaz de reconocer un conjunto completo de letras mayúsculas y minúsculas, números y signos de puntuación.

Un esquema de este tipo requiere un planteamiento más radical que los utilizados anteriormente, pero resulta perfectamente posible cumplir todas las condiciones y el fundamento del programa que figura a continuación es precisamente un sistema de este tipo.

SISTEMA DE COMPRESION

Este sistema compresor de textos utiliza algunos principios tomados del mundo de la criptografía, disciplina que se ocupa del estudio de los códigos y los mensajes cifrados. Dichos

principios nos pueden permitir saber algo acerca de la estructura del texto que se somete a compresión.

Cuando se intenta romper un código, es muy útil conocer la frecuencia con que aparecen las distintas letras en el idioma español, ya que ello permite que el descifrador identifique ciertos grupos que intervienen en los mensajes cifrados. Durante mucho tiempo los criptógrafos han trabajado afanosamente sumergiéndose en largos textos y contando el número de veces que se presenta cada letra, cada par de letras o incluso cada palabra.

El compresor de texto que veremos en **INPUT** está basado en la frecuencia de aparición de las letras sueltas y, en determinados casos, de pares de letras. Se asigna números en binario a cada una de las letras del alfabeto; a las letras más frecuentes se les asignan los números más bajos para ahorrar y a las menos frecuentes se les asignan los números binarios más altos.

Una consecuencia importante de utilizar este tipo de codificación es que, al realizar la prueba del *software* de compresión con grupos aleatorios de letras, sin significado alguno, es muy posible que resulte poca o ninguna compresión. Si por el contrario empleas un texto en español te encontrarás que la compresión es bastante grande. También es interesante observar que si realizas por ejemplo la compresión de un texto en inglés o en francés, el efecto de compresión obtenido es considerablemente menor, debido a que las frecuencias de aparición de cada letra son diferentes en cada idioma, por lo que se requeriría un esquema de codificación diferente.

Además de los códigos de las letras individuales, el compresor también se ocupa de las dos letras más frecuentes que pueden seguir a cada letra. En la tabla 1 que ofrecemos en el próximo capítulo tienes un resumen de los códigos del idioma inglés, en el que fue originalmente desarrollado este programa. Más adelante mostraremos cómo adaptarlo en tu juego particular o en idioma español, por ejemplo.

Observarás que por ejemplo, la letra que sigue a la T con mayor frecuencia es la H, y a continuación la I. Para estas combinaciones de números muy comunes, también se han adoptado por razones de economía códigos especiales.

Cuando el programa compresor está realizando la codificación de un texto, recuerda cuál era la última letra y comprueba si la letra siguiente es la primera o la segunda más frecuentes. Si se trata de una de dichas letras más frecuentes, utiliza el código de la parte superior de la columna. Si por el contrario la letra siguiente no es una de las más comunes, se utiliza el código ordinario de la letra que se presenta sola. Este principio se aplica durante todo el proceso de codificación. Para cada nueva letra o carácter, lo primero que hace la máquina es comprobar si se trata de uno de los dos caracteres más comunes que siguen al anterior.



ACCESO AL COMPRESOR DE TEXTOS

El compresor de textos ha sido escrito en código máquina para acelerar la codificación y decodificación del material a procesar.

El listado se presenta tanto en forma hexadecimal como en lenguaje ensamblador. En esta parte de nuestro coleccionable te presentamos un listado en lenguaje ensamblador correspondiente a la parte del programa encargada de la codificación. En el próximo capítulo del coleccionable veremos la parte de decodificación del listado en ensamblador, así como el listado en forma hexadecimal que completan el programa.

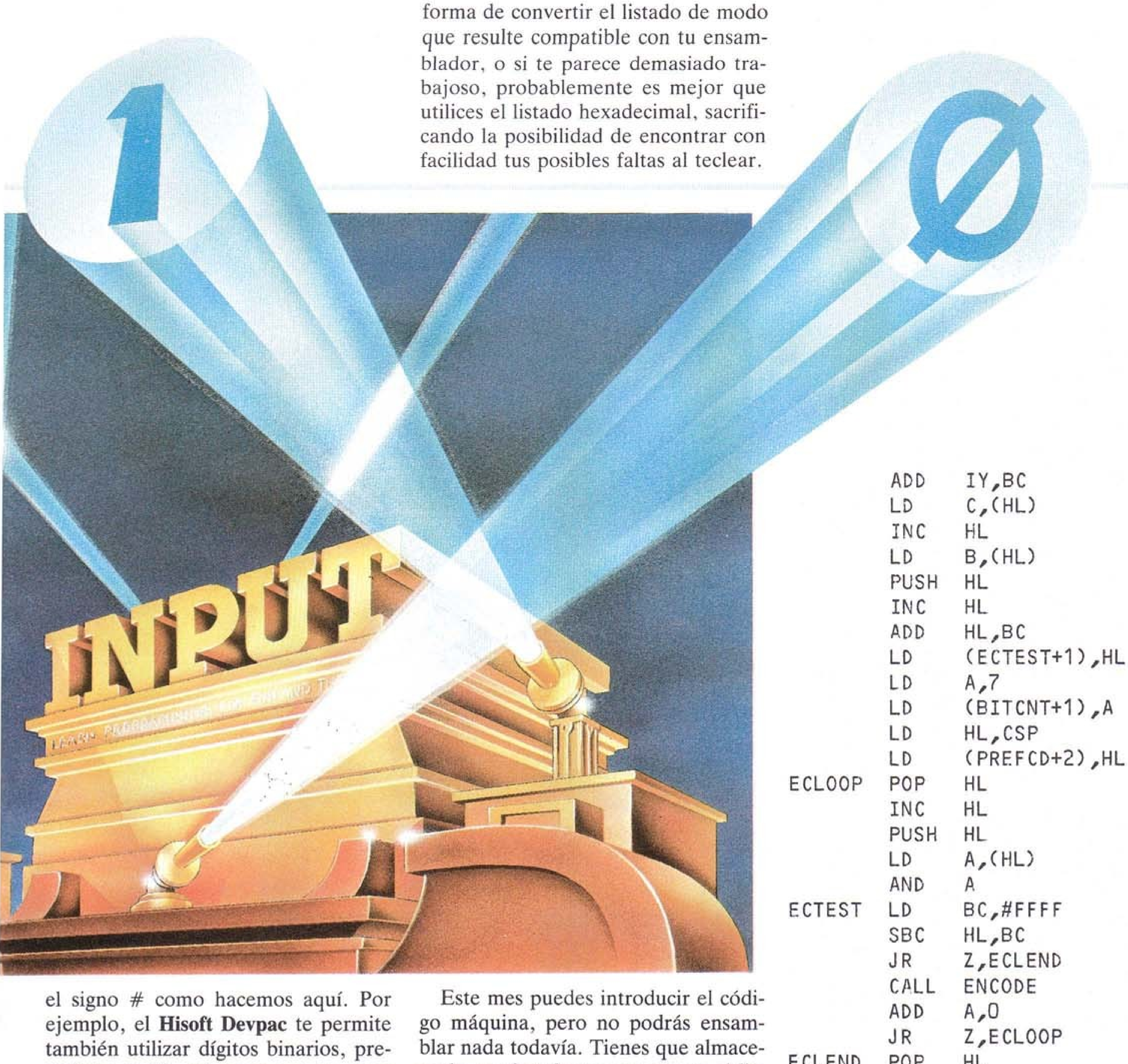
Necesitarás un ensamblador comercial para utilizarlo con el listado de ensamblador, o en caso contrario tendrás que esperar a tener el listado de código máquina.

Si no dispones de un ensamblador comercial, tienes que introducir en tu máquina un corto programa en BASIC, ejecutarlo e introducir todo el código hexadecimal; el programa POKEará los valores hexadecimales en las direcciones de memoria adecuadas. Pero ya nos ocuparemos más detalladamente de esto la próxima vez.

Hemos escrito el programa compresor de textos utilizando un ensamblador estandar de los varios que hay en las tiendas por lo que podrías encontrarte con que en el siguiente listado hay algo que tu ensamblador no reconoce. Si utilizas uno diferente al nuestro, comprueba que te permite evaluar expresiones aritméticas; si no te lo permite, probablemente lo más fácil que puedes hacer es introducir el listado hexadecimal que veremos en el siguiente número de **INPUT**.

Otra de las variantes típicas de unos ensambladores a otros es el uso de la letra h a continuación de un número para indicar que se trata de un número en hexadecimal, en vez de utilizar

forma de convertir el listado de modo que resulte compatible con tu ensamblador, o si te parece demasiado trabajoso, probablemente es mejor que utilices el listado hexadecimal, sacrificando la posibilidad de encontrar con facilidad tus posibles faltas al teclear.



el signo # como hacemos aquí. Por ejemplo, el **Hisoft Devpac** te permite también utilizar dígitos binarios, precedidos de %, mientras que en otros ensambladores tendrás que calculártelos tú a mano, ya que es probable que no entiendan los números en binario. Hay algunos ensambladores que no soportan la instrucción EQU, por lo que al intentar omitirla por completo lo que resultaría sería una línea con la etiqueta y nada más. Finalmente, hay algunos ensambladores que reconocen una forma no estándar de la instrucción ADD; ensaya por ejemplo la sustitución de ADD A,8 por ADD 8.

Si no te sientes muy seguro sobre la

Este mes puedes introducir el código máquina, pero no podrás ensamblar nada todavía. Tienes que almacenarlo en cinta hasta que tengas el listado completo.

```
        ORG     64600
ESTART  LD      HL,11
        LD      (MARRY+1),HL
        RET
ECODES  PUSH    IX
        PUSH    IY
        CALL    ESETUP
MARRY   LD      BC,#FFFF
        LD      (IY+8),C
        LD      (IY+9),B
        PUSH    IY
        ADD     IY,BC
        LD      C,(HL)
        INC     HL
        LD      B,(HL)
        PUSH    HL
        INC     HL
        ADD     HL,BC
        LD      (ECTEST+1),HL
        LD      A,7
        LD      (BITCNT+1),A
        LD      HL,CSP
        LD      (PREFCD+2),HL
ECLOOP  POP     HL
        INC     HL
        PUSH    HL
        LD      A,(HL)
        AND     A
ECTEST  LD      BC,#FFFF
        SBC     HL,BC
        JR      Z,ECLEND
        CALL    ENCODE
        ADD     A,0
        JR      Z,ECLOOP
ECLEND  POP     HL
        LD      A,0
        CALL    ENCODE
        INC     IY
        PUSH    IY
        POP     HL
        POP     BC
        LD      A,(BITCNT+1)
        ADD     A,249
        SBC     HL,BC
        LD      (MARRY+1),HL
        POP     IY
        POP     IX
        RET
```

```
ENCODE  LD    B,CLAST-CODE+1
        CP    "@"
        JR    C,LAKE
        XOR   #20
LAKE    LD    C,A
        LD    HL,CLAST
TRYC    SUB   (HL)
        JR    Z,MATCH
        CP    #20
        JR    NZ,NOUPP
        LD    A,"^"+#20
REDO    PUSH  HL
        PUSH  BC
        CALL  ENCODE
        POP   BC
        POP   HL
        JR    MATCH
NOUP    CP    #E0
        JR    NZ,NOLOW
        LD    A,"-"+#20
        JR    REDO
NOLOW   LD    A,C
        DEC   HL
        DJNZ  TRYC
        RET
MATCH   LD    A,B
PREFCD  LD    IX,#FFFF
BITCNT  LD    E,#FF
        DEC   E
        DEC   E
        DEC   A
        JR    Z,SPACEE
        CP    (IX+FIRST-CODE)
        JR    NZ,NOTFIR
        LD    A,0
        JR    TOP
NOTFIR  DEC   E
        CP    (IX+SECOND-
              CODE)
        JR    NZ,LONG
        LD    A,%010*32
        JR    TOP
SPACEE  LD    A,%011*32
        DEC   E
        JR    TOP
LONG    DEC   E
        DEC   E
        DEC   E
        DEC   E
        DEC   E
        ADD   A,255+CODE-
              CLAST
        CP    255+CEIGHT-
              CLAST
        JR    NC,TOP
        INC   E
        SLA   A
        ADD   A,CLAST-CEIGHT
              +1
        CP    255+CSEVEN-
              CLAST
              +CSEVEN-CEIGTH
        JR    NC,TOP
        INC   E
        SLA   A
        ADD   A,CLAST-CSEVEN
              +CEIGHT-CSEVEN
              +1
        CP    255+CSIX-
              CSEVEN+CSIX-
              CSEVEN+CSIX
              -CEIGHT+CSIX
              -CLAST
        JR    NC,TOP
        INC   E
        SLA   A
        ADD   A,CLAST-CSIX+
              CEIGHT-CSIX+
              CSEVEN-CSIX+
              CSEVEN-CSIX+1
TOP     LD    C,A
        LD    A,(HL)
        CP    "^"
        JR    Z,TOPOUT
        LD    A,(IX)
        CP    "-"
        JR    NZ,HOLD
        LD    HL,CPUN
HOLD    LD    (PREFCD+2),HL
TOPOUT  LD    A,C
TOPACT  LD    BC,(BITCNT+1)
        LD    B,C
        INC   B
JAR     RLCA
        DJNZ  JAR
        LD    IX,LO
        ADD   IX,BC
        LD    B,A
        AND   (IX+1)
        OR    (IY)
        LD    (IY),A
        LD    A,B
        AND   (IX+UP-LO+1)
        LD    (IY+1),A
        LD    A,E
        CP    %10000000
        JR    C,DONE
        ADD   A,8
        INC   IY
DONE    LD    (BITCNT+1),A
        LD    A,0
        RET
CODE    EQU   $
CSP     DEFB  " "
CA      DEFB  "A"
CS      DEFB  "S"
CO      DEFB  "O"
CT      DEFB  "T"
CR      DEFB  "R"
CI      DEFB  "I"
CL      DEFB  "L"
CE      DEFB  "E"
        DEFB  "C"
CSIX    EQU   $
CUP     DEFB  "^"
CAR     DEFB  "-"
CU      DEFB  "U"
        DEFB  "M"
Cp      DEFB  "P"
        DEFB  "W"
        DEFB  "Y"
CN      DEFB  "N"
        DEFB  "B"
        DEFB  "G"
CSEVEN  EQU   $
CD      DEFB  "D"
CF      DEFB  "F"
        DEFB  "V"
CH      DEFB  "H"
CEIGHT  EQU   $
        DEFB  "K"
        DEFB  "Q"
CX      DEFB  "X"
        DEFB  "J"
        DEFB  "Z"
        DEFB  "["
CPO     DEFB  "\"
CLAST   DEFB  "]"
CPUN    EQU   $
```

Esta aplicación se divide en tres capitulos. En el primero abordamos el tema del compresor, en el segundo tratamos del descompresor y en el tercero incorporaremos un programa que nos permitirá su utilizacion desde el BASIC.

DETECCION DE OBJETOS EN LA PANTALLA

- DETECCION DE FORMAS EXTRAÑAS
- COMPROBACIONES DE COLOR
- EL COMANDO ATTR
- LA PELOTA QUE REBOTA
- DETECCION DE LOS CHOQUES

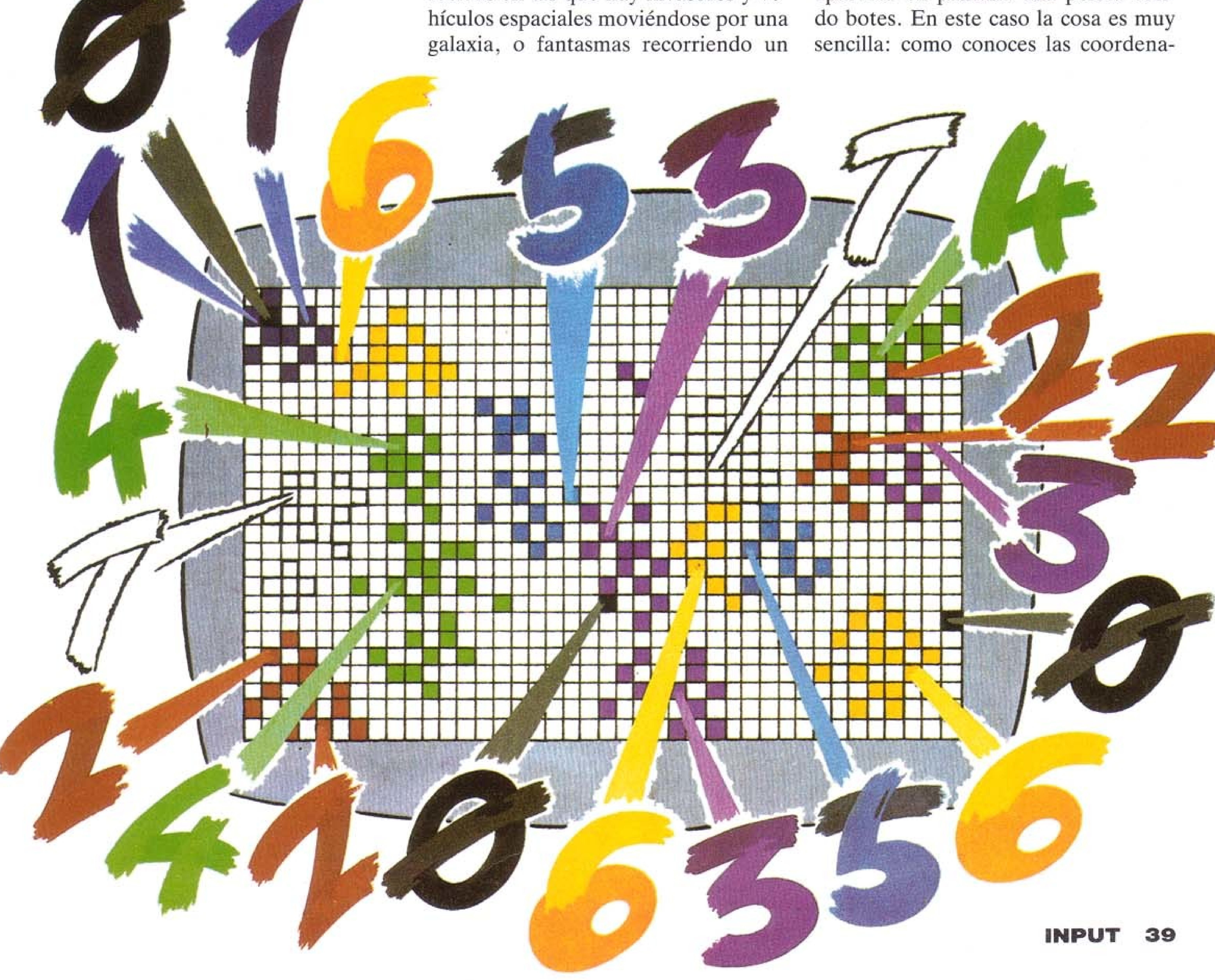
Tu ordenador puede seguir a los gráficos utilizando determinados comandos, que le permiten observar su propia pantalla. Esto resulta especialmente útil en los juegos en los que se producen colisiones

En las situaciones en que hayas creado una figura detallada sobre la pantalla, ¿cómo puedes asegurarte de que el siguiente objeto que añadas a la imagen no coincidirá con algo que ya había allí antes?

Una respuesta obvia es que seas cuidadoso y sistemático al llevar la cuenta de las zonas de la pantalla que ya han sido utilizadas. Pero esto puede llegar a resultar enormemente complicado, y hay circunstancias en que puede ser casi imposible, como es el caso de los gráficos móviles. Y sin embargo éste es precisamente el tipo de problemas con el que te encuentras al escribir programas de juegos y cosas parecidas en las que hay invasores y vehículos espaciales moviéndose por una galaxia, o fantasmas recorriendo un laberinto. En tales casos tienes que asegurarte de que no se impriman dos gráficos en el mismo sitio, o de qué sucederá si se produce la colisión y, por ejemplo, se desintegre la nave espacial.

DETECCION DE FORMAS EXTRAÑAS

Supongamos por ejemplo que deseas escribir un programa que haga aparecer en pantalla una pelota dando botes. En este caso la cosa es muy sencilla: como conoces las coordena-



das de los cuatro lados de la pantalla, puedes incluir en tu programa cuatro condiciones IF ... THEN para comprobar si la pelota ha llegado o no a uno de los cuatro lados. Ello se hace comparando las coordenadas de la pelota con las coordenadas conocidas de los lados. Pero ¿qué sucede cuando lo que quieres comprobar es si la pelota ha golpeado contra un objeto de forma más extraña, por ejemplo circular?

Podrías utilizar el mismo método, disponiendo unas cuantas comprobaciones del tipo IF ... THEN que contengan los detalles de las coordenadas del círculo para ver en qué puntos golpea la pelota con los lados. Pero debido a que la forma curva es especialmente compleja, se necesitaría un número de comprobaciones muy grande. Y ya sabes que el ordenador necesita mucho tiempo para ejecutar cada comprobación IF ... THEN. Un programa escrito de esta forma resultaría extremadamente lento, y el movimiento resultante sobre la pantalla se produciría a base de tirones o sacudidas.

Existe un límite para las velocidades que es posible obtener utilizando el BASIC. Pero tu ordenador tiene un comando que te permite detectar la presencia de cualquier objeto sobre la pantalla con mucha mayor rapidez que la comprobación de las coordenadas. Esto es posible aunque no se conozca su posición.

EL COLOR POR MEDIO DE NUMEROS

El comando ATTR del **Spectrum** te devuelve como respuesta el color especificado en cada carácter cuadrado. Esto significa que puedes detectar la presencia de cualquier objeto, sin más que especificar su color y comprobando todas las posiciones de pantalla. Así, en el ejemplo anterior, con un círculo rojo resultaría el número de código del color rojo cada vez que se presentara. Comprobando únicamente este código el programa podría hacer que la pelota rebotara hacia afuera del círculo, o cualquier otra cosa.

Realmente el **Spectrum** da un número entre 0 y 225, que tiene en cuenta todos los atributos de cada carácter cuadrado: los colores del papel y de la tinta, si se presenta con brillo o no y si hay o no intermitencia.

En cuanto a la sintaxis de este comando, ATTR ha de ir seguido de las coordenadas de pantalla del carácter cuyo color quieres averiguar, colocadas entre paréntesis. Podrías escribir pues algo como esto:

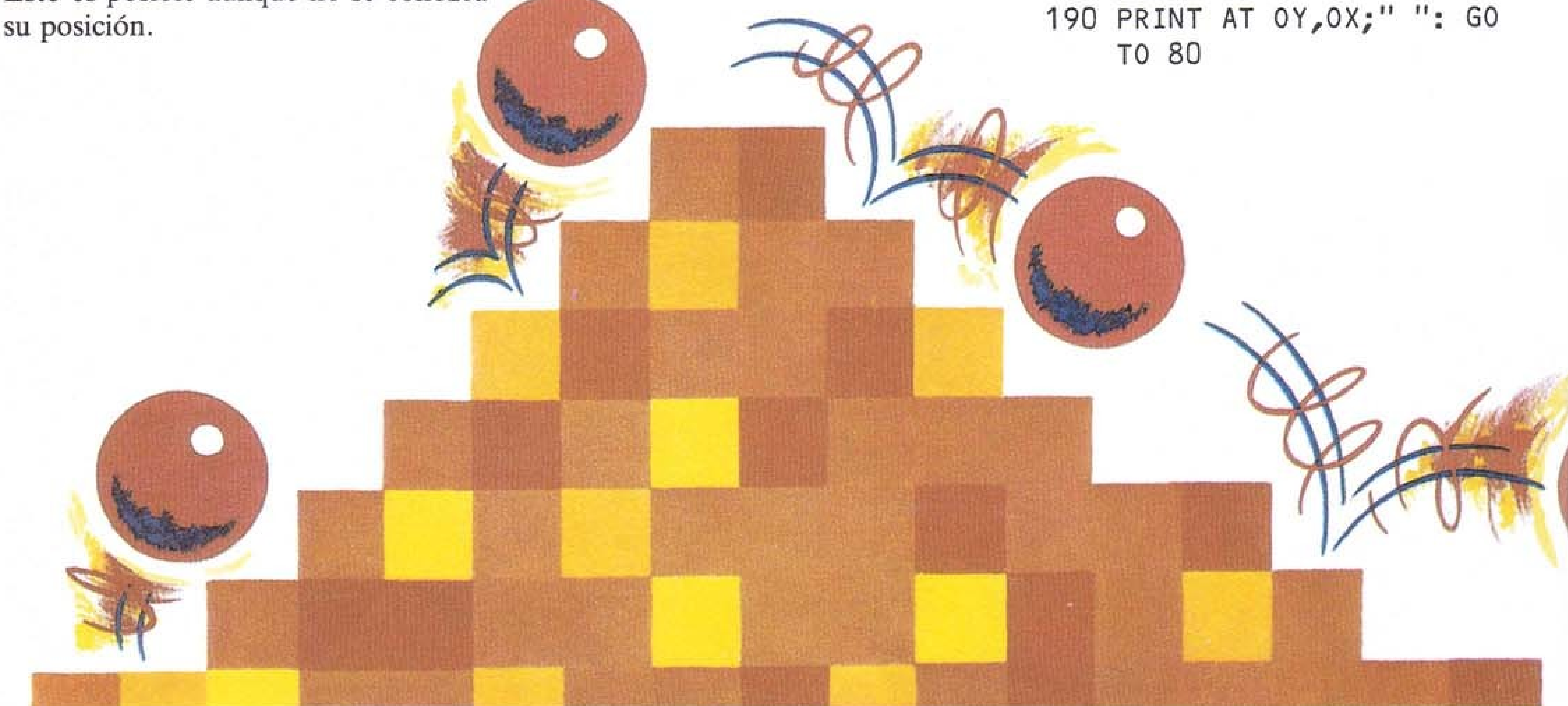
```
PRINT ATTR(10,10)
```

Por desgracia, no puedes invertir el proceso y cambiar los colores de cada carácter cuadrado. Sin embargo, sí que puedes POKEar diferentes valores en el fichero de atributos de tu ordenador.

Teclea y ejecuta ahora el siguiente programa que es un ejemplo de utilización del comando ATTR.

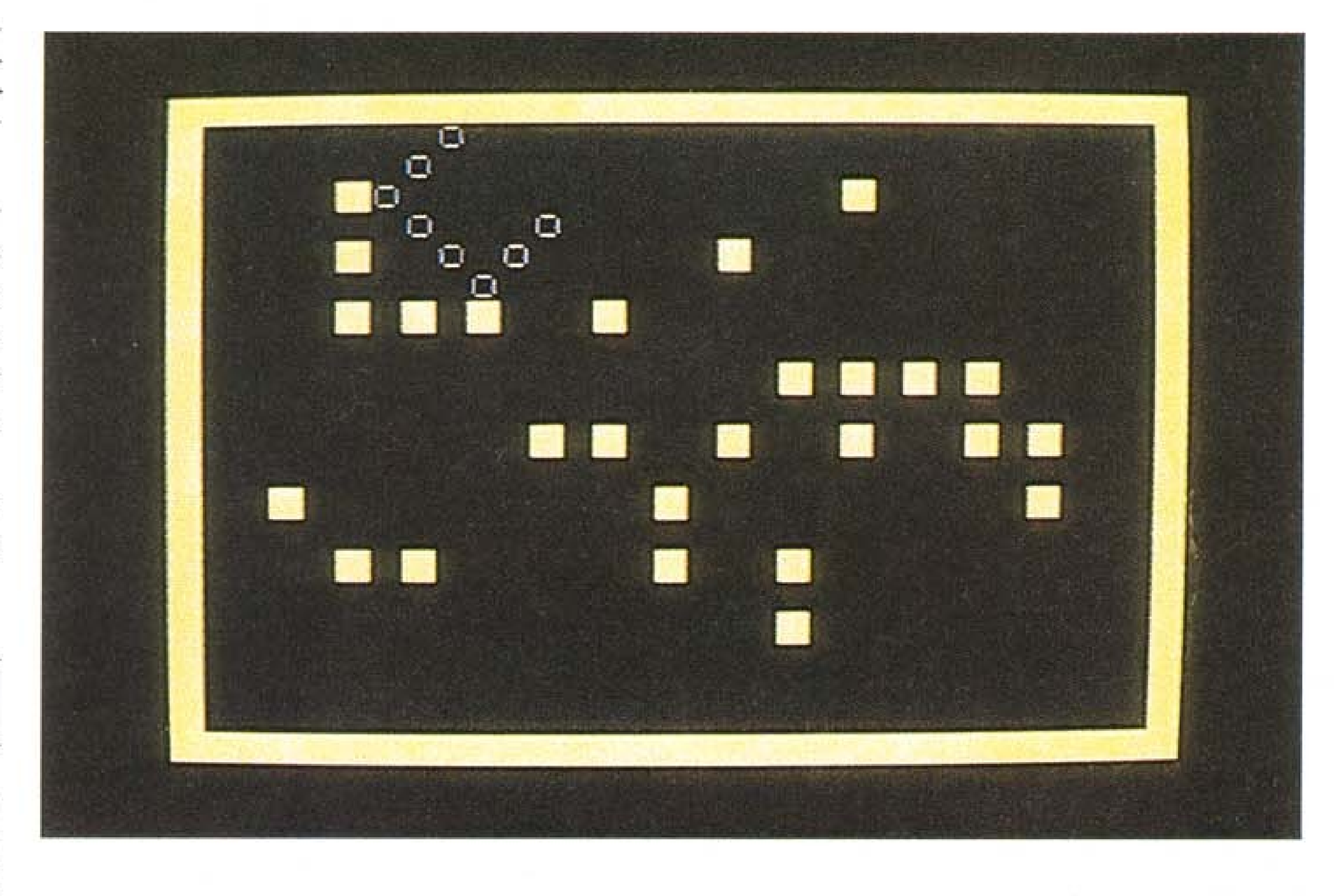
Teclea

```
10 BORDER 0: PAPER 0: INK 9
15 CLS
20 FOR N=22528 TO 22559:POKE
   N,48: POKE N+672,48: NEXT
   N
30 FOR N=22560 TO 23168 STEP
   32: POKE N,48: POKE N+31,
   48: NEXT N
40 FOR N=1 TO 30: PRINT
   PAPER 6;AT INT (RND*8)*2+
   3,INT (RND*13)*2+3;" ":
   NEXT N
50 LET X=15: LET Y=10: LET
   XV=-1: LET YV=1
80 PRINT AT Y,X;"O": LET OY=
   Y: LET OX=X
90 LET X=X+XV: LET Y=Y+YV
140 IF ATTR (Y,X-1)=48 OR
    ATTR (Y,X+1)=48 THEN LET
    XV=-XV
145 IF ATTR (Y-1,X)=48 OR
    ATTR (Y+1,X)=48 THEN LET
    YV=-YV
190 PRINT AT OY,OX;" ": GO
    TO 80
```



La imagen que aparece en el **Spectrum** consta de una serie de espacios amarillos presentados aleatoriamente dentro de un recinto amarillo. Al hacer correr el programa, el ordenador empieza a mover una pelota con una dirección inicial aleatoria.

Cuando la pelota llega al borde de uno de los bloques, rebota y toma una dirección diferente.

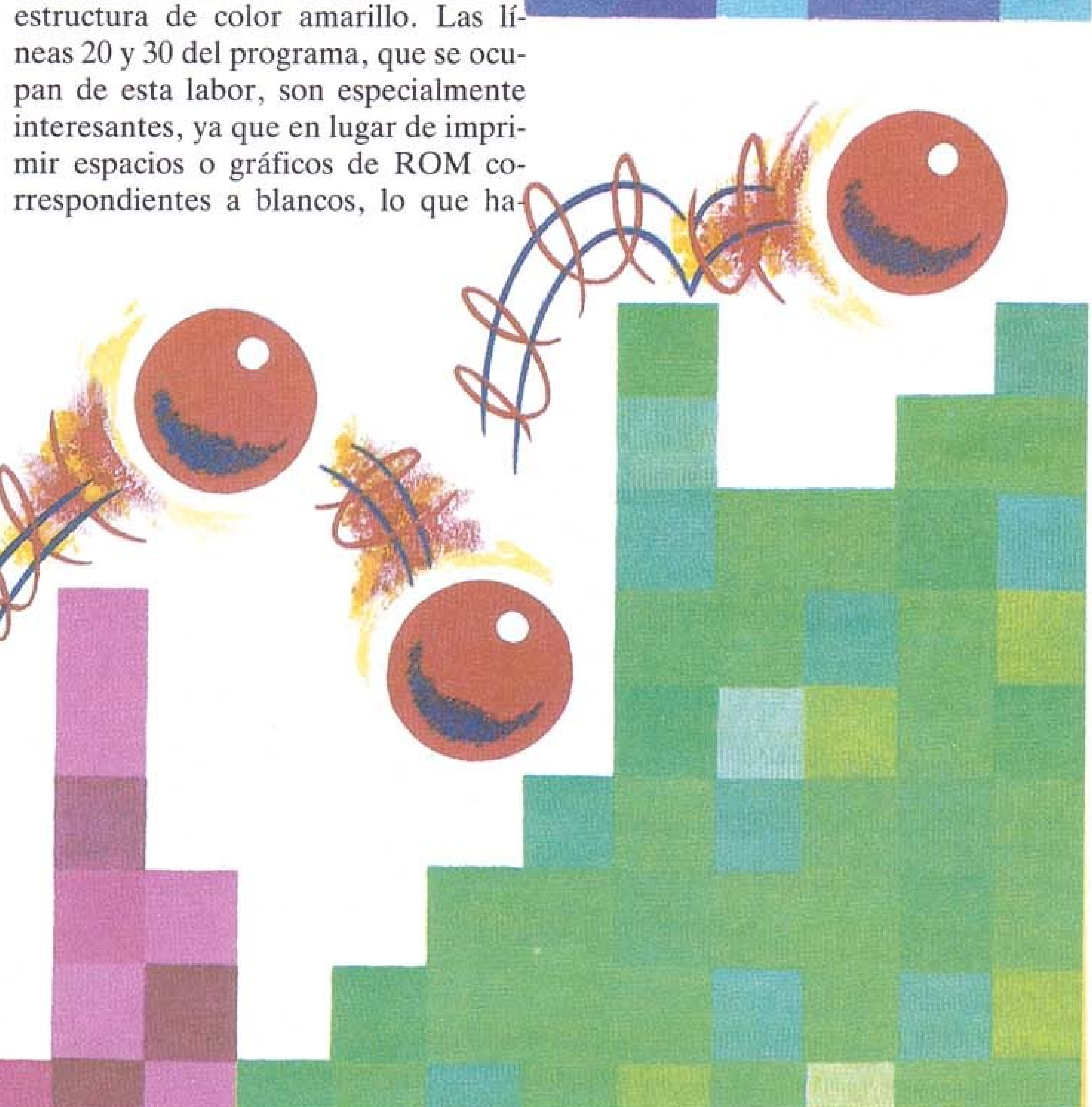


DETECCION DE LAS COLISIONES

Las comprobaciones para detectar el instante en que la pelota choca con el borde de uno de los cuadrados, se hacen por medio del comando ATTR. Como los cuadrados se imprimen aleatoriamente, la única alternativa para el seguimiento sería asignar una variable para la coordenada x y otra para la coordenada y de cada cuadrado. De esta forma, al utilizar dos variables para cada cuadrado, aparte del hecho de utilizar una gran cantidad de memoria, se necesitarìa una cantidad enorme de comprobaciones IF ... THEN. Estas comprobaciones harían que lo que ahora es un programa razonablemente rápido, se convirtiera en un programa insoportablemente lento.

Sería muy sencillo para tí utilizar comprobaciones del tipo IF ... THEN para detectar las colisiones de la pelota con los bordes, ya que sus coordenadas se definen con mucha facilidad. Pero una vez que hayas utilizado el comando ATTR para comprobar si la pelota ha chocado o no contra un bloque amarillo, lo suyo es utilizarlo también para las colisiones con los bordes, especialmente en este caso en que el color es el mismo que el de los bloques.

El programa empieza por establecer los colores de la pantalla y creando la estructura de color amarillo. Las líneas 20 y 30 del programa, que se ocupan de esta labor, son especialmente interesantes, ya que en lugar de imprimir espacios o gráficos de ROM correspondientes a blancos, lo que hacen es POKEar valores en el fichero de atributos. Hay dos bucles FOR ... NEXT que recorren las direcciones de memoria que contienen los atributos correspondientes a los caracteres cuadrados del borde de la pantalla. En ca-



da dirección se POKEa el valor 48, que es el código correspondiente a papel amarillo con tinta negra y sin activar el brillo ni la intermitencia.

El programa continúa imprimiendo bloques de forma aleatoria (esta vez se trata de espacios). Los números aleatorios que controlan las posiciones en que se ejecutan los PRINT, se eligen de forma que la zona de trabajo se encuentre situada alejada por lo menos dos caracteres del borde de la pantalla; de esta forma se evita que se sitúen caracteres aleatorios sobre el mismo.

En la línea 50 se inicializan unas cuantas variables: la x y la y, correspondientes a la posición inicial de la pelota y las xv e yv, que corresponden a las componentes de la dirección inicial de la pelota según los ejes horizontal y vertical.

Después de hacer esto el ordenador salta a la línea 80, encargándose de la presentación de la pelota y definiendo otras dos variables más: ox y oy, que corresponden a los valores antiguos de x e y. Estos valores se utilizan para borrar la pelota después de que x e y han sido modificados por los cálculos subsiguientes, destinados a determinar el movimiento de la pelota.

Los cálculos se hacen en la línea 90. Lo que se hace es sumar los vectores de dirección (o la distancia viajada en cada dirección), determinados por xv e yv.

USO DE ATTR

Las importantes líneas 140 y 145 son las que se ocupan de comprobar los colores de los cuatro cuadrados que rodean a la pelota.

Como puedes ver, la función ATTR se escribe así:

ATTR (coordenada y, coordenada x)

Pero, al igual que PEEK, ATTR no es un comando directo. En consecuencia no puedes hacer que tu ordenador ejecute directamente ATTR como podrías hacer por ejemplo con PRINT o LOAD. Por eso va contenido dentro de una instrucción, que en este caso es la IF ... THEN.

La sentencia LET que figura a continuación de las condiciones de estas dos líneas, lo que hace es invertir simplemente el vector velocidad cuando el color del siguiente cuadrado al que va a llegar la pelota es el amarillo (ATTR = 48 corresponde a un carácter cuadrado amarillo).

El resultado que se obtiene en la práctica es que la pelota rebote con un angulo como el que resultaría si la pelota «chocase» con el cuadrado. En la dirección y, la pelota tiene que venir desde arriba o desde abajo en el momento de la colisión, mientras que en la dirección x, ha de llegar desde uno de los lados del obstáculo.Si se invierten ambos vectores al mismo tiempo, la pelota se irá por el mismo camino por el que venía. Por ello las comprobaciones sólo afectan a la componente de la velocidad en la dirección del choque. Su velocidad en otras direcciones (en las que no se produce colisión) sigue siendo la misma. Según esto, si se detecta que la pelota está a punto de chocar con un bloque amarillo que hay por debajo de ella, empezará a moverse hacia arriba, pero la componente horizontal de su velocidad (dirección x) no se verá afectada.

Después de dos comprobaciones, el ordenador borra la imagen de la pelota antigua, escribiendo sobre ella con un espacio en blanco (línea 190) y volviéndose de nuevo a la línea 80 para continuar.

Sería muy sencillo añadir unas cuantas líneas al programa anterior para convertirlo en un juego. En futuros artículos de **INPUT** utilizaremos este comando en una amplia variedad de rutinas de juegos.

EL ZOCO DE INPUT

Todo se compra y se vende. Los antiguos zocos fueron lugares destinados a todo tipo de transacciones. INPUT también tiene el suyo. Vuestras operaciones de compra, cambio o venta serán publicadas en esta sección, pero dos son las limitaciones que imponemos:

a) La propuesta tendrá que ver con la microinformática.

b) Nos reservamos el derecho de no publicar aquellos insertos de los que se sospeche un trasfondo lucrativo.

Ahora un ruego. Tratar de resumir al máximo el texto; escribir casi como un telegrama siendo claros y concisos.

Envía tu mensaje a:

INPUT SINCLAIR-ZOCO
c./ Alberto Alcocer, 46
28016 **MADRID**


MICRO-1
C/ Duque de Sesto, 50. 28009 Madrid
Tel.: (91) 275 96 16/274 53 80
(Metro O'Donell o Goya)
el IVA lo pag
MICRO-1
1.395 ptas.
1.695 ptas.
1.695 ptas.
QUICK SHOT I+INTERFACE
2.695 PTAS.
QUICK SHOT II+INTERFACE
2.995 PTAS.
QUICK SHOT V+INTERFACE
2.995 PTAS.
NECESITAMOS DISTRIBUIDORES ¡¡GRANDES DESCUENTOS!!
DIPROIMSA
DISTR. de PRODUCTOS INFORMATICOS M., s.a.
C/ GALATEA, 25. 28042 MADRID
TF. 742 20 19 - 274 53 80
Recorta o copia este cupón y envíalo a:
MICRO 1. C/ Duque de Sesto, 50. 28009 MADRID. Tf.: 275 96 16.
NOMBRE
APELLIDOS
CALLE
C. POSTAL
CANTIDAD
DESCRIPCION
PTAS.
PROVINCIA
¡SIN GASTOS DE ENVIO!

Recetario práctico de C. M.

DIBUJANDO EN CODIGO MAQUINA

Continuamos tratando algunas rutinas en código máquina que nos serán de gran utilidad a la hora de realizar nuestros propios programas.



UTILIZACION DE DRAW

La rutina consta de varias partes:

- Posicionamiento del punto de origen de la recta mediante la rutina PLOT ($22E5), semejante a la instrucción correspondiente en BASIC. Cargaremos el valor de la coordenada «x» en el registro simple C y el de la «y» en el B, o directamente ambos en el BC (operando adecuadamente). Una vez hecho esto, haremos CALL #22E5.

- Utilización de la rutina DRAW desde el código máquina. Aunque existen variantes de esta rutina en la ROM, vamos a usar la ubicada en la posición $24BA, por entenderse que es la de más fácil y frecuente manejo. En el registro C cargamos el valor del desplazamiento horizontal, en el B el del desplazameinto vertical, posicionando así el punto que será unido con el elegido mediante PLOT. Los signos de la línea a dibujar se introduce en el registro DE, siendo #$01 positivo, y #$FF negativo. Ej.: LD DE,#$FF01 indica un desplazamiento horizontal positivo, y vertical negativo (el registro E con el signo de la «x», el D con el de la «y»).

- También llamamos la atención del lector sobre el uso de la línea 450 como método de almacenamiento de valores a modo de «variables». Se reserva un espacio con DEFB que a la vez que asigna un valor inicial puede servir para almacenar otros resultados.

- Las demás instrucciones han sido comentadas en el número anterior (borrado de pantalla, apertura de canal...), únicamente comentar la necesidad de conservar el valor del registro HL para poder volver al BASIC, ya que el uso de DRAW o CIRCLE desde C/M modifican su valor (líneas 100,110,130 y 410,420,430).

A modo de «receta» se puede resumir la utilización del DRAW desde el C/M como se indica en la tabla número 1.

TABLA 1

```
EXX           ;Conmuta al juego alternativo de
               registros
PUSH HL       ;Guarda el H'L'
EXX           ;Pasa de nuevo al set de registros
               normal
LD BC,256*(valor de "y")+(valor de "x")
LD DE,#yyxx   (signo de y, signo de x, en hex.
               #01 o #FF)
CALL #24BA
EXX           ;Registros alternativos
POP HL        ;Recupera el H'L'
EXX           ;Pasa al set normal
```


Y a la venta en: GALERIAS PRECIADOS, PRYCA, CONTINENTE y establecimientos especializados.
MAX HEADROOM
Lo
que
yo
quisiera
saber
es . . .
Disponible
en - CBM 64
SPECTRUM - 48/128 K
y AMSTRAD.
Porqué
alguien
querrá
jugar
con
algo
diferente?
QUICKSILVA
Los juegos más poderosos del universo
MIND GAMES ESPAÑA S.A.
MARIANO CUBI, 4 ENTLO.
TEL. 218 34 00
08006 BARCELONA
© Chrysalis Visual Programming Ltd.
Deseo recibir los juegos que a continuación especifico, comprometiéndome al pago del importe de los mismos.
Nombre
Dirección
Teléfono
Firma:
MAX HEADROOM
SISTEMA
CANTIDAD
Contrarreembolso.
Adjunto Talón.
Giro Postal.
Deseo recibir información de sus programas en:
MSX
AMSTRAD

RUTINA 1

```
10         ORG  50000       ;Lugar de colocacion del C/M
20         ENT  50000       ;Direccion de ejecucion (con el GENS3)
30         LD   HL,23693    ;Variable del sistema ATTR P (colores)
40         LD   (HL),112    ;Tinta 0, papel 6, brillo 1, flash 0
50         CALL #0D6B       ;borrado de pantalla
60         LD   A,2         ;Carga 2 en el acumulador para abrir el
70         CALL #1601       ;canal de pantalla
80         LD   BC,0        ;Se carga B con 0, C con 0 y se ejecuta
90         CALL #22E5       ;la rutina PLOT (PLOT 0,0)
100        EXX              ;Estas tres lineas guardan el registro
110        PUSH HL          ;HL para poder recuperar su valor al
120        EXX              ;termino de la rutina
130 BUCLE  LD   A,(LADO)    ;Tomamos el valor de LADO (LET A=LADO)
140        LD   C,0         ;Hace LET C=0 (coordenada x)
150        LD   B,A         ;Hace LET B=A (coordenada y)
160        LD   DE,#0101    ;Signo de x=+ , signo de y=+
170        CALL #24BA       ;Ejecuta DRAW (DRAW 0,A)
180        LD   A,(LADO)    ;El proceso se va a repetir 3 veces mas
190        LD   C,A         ;para acabar el cuadrado iniciado
200        LD   B,0         ;Las lineas 180,190,200,210,220, son
210        LD   DE,#0101    ;equivalentes a DRAW A,0
220        CALL #24BA       ;
230        LD   A,(LADO)    ;Las lineas 230,240,250,260,270, son
240        LD   B,A         ;equivalentes a DRAW 0,-A
250        LD   C,0         ;
260        LD   DE,#FF01    ;Signo de x=+ , signo de y=-
270        CALL #24BA       ;
280        LD   A,(LADO)    ;Las lineas 280,290,300,310,320, son
290        LD   C,A         ;equivalentes a DRAW -A,0
300        LD   B,0         ;
310        LD   DE,#01FF    ;Signo de x=- , signo de y=+
320        CALL #24BA       ;
330        LD   A,(LADO)    ;Carga en A el valor que este en LADO
340        DEC  A           ;Disminuye en 2 unidades el valor de A,
350        DEC  A           ;siendo el nuevo valor de LADO que se
360        LD   (LADO),A    ;introduce con 360. Equivale a STEP -2
370        CP   1           ;Cuando el LADO llegue a 1 se activa el
380        JR   NZ,BUCLE    ;flag Z. Si no ocurre, se vuelve a BUCLE
390        LD   A,175       ;Una vez acabada la rutina se restauran
400        LD   (LADO),A    ;los valores para poder volver a rodar
410        EXX              ;Para poder volver sin problemas al
420        POP  HL          ;Basic recupera el valor de HL guardado
430        EXX              ;al principio de la rutina
440        RET              ;RETorno al BASIC
450 LADO   DEFB 175         ;Contenido inicial de LADO
```

La rutina quiere conseguir el mismo efecto que las líneas en BASIC que mostramos a continuación:

```
10 PAPER 6: INK 0: BRIGHT 1:
   CLS
20 PLOT 0,0: FOR N=175 TO 1
STEP -2: DRAW 0,N: DRAW N
,0: DRAW 0,-N: DRAW -N,0:
NEXT N
```

Que en código máquina queda como se indica en la RUTINA 1.

(Ver cuadro arriba.)

Para los que no dispongan de un programa ensamblador adjuntamos el siguiente cargador BASIC de la rutina:

```
10 CLEAR 49999: FOR N=50000
   TO 50091: READ A: POKE N,
```

RUTINA 2

```
10          ORG  60000       ;Direccion de colocacion y ejecucion
20          ENT  60000       ;La rutina es reubicable
30          CALL #0D6B       ;Borra la pantalla
40          LD   A,2         ;Abre el canal 2 para poder escribir
50          CALL #1601       ;en pantalla
60          EXX              ;Guarda el valor de HL para poder
70          PUSH HL          ;regresar al BASIC
80          EXX              ;
90  BUCLE   LD   A,(VALOR)   ;Hace LET A=(contenido de VALOR)
100         CALL #2D28       ;Lo mete en stack como "x" del centro
110         LD   A,(VALOR)   ;Carga A con el contenido de VALOR
120         CALL #2D28       ;Lo mete en stack como "y" del centro
130         LD   A,(VALOR)   ;Asi los 3 parametros son iguales.Este
140         CALL #2D28       ;valor pasa al stack como "radio"
150         CALL #232D       ;Ejecuta la rutina CIRCLE
160         LD   A,(VALOR)   ;Pasa el contenido de VALOR al A
170         INC  A           ;Incrementa el valor cargado en 2. Es
180         INC  A           ;como STEP 2, y pasar al siguiente
190         LD   (VALOR),A   ;El valor incrementado pasa a VALOR
200         CP   88          ;Si el contenido guardado en VALOR es
210         JR   NZ,BUCLE    ;dist. de 88 va a BUCLE
220         LD   A,0         ;Cuando llega a 88 se acaba la rutina
230         LD   (VALOR),A   ;y estas lineas ponen a 0 VALOR
240         EXX              ;Recupera el valor de HL guardado en
250         POP  HL          ;las primeras lineas del programa
260         EXX              ;
270         RET              ;Retorno al BASIC
280 VALOR   DEFB 0           ;Lugar donde va almacenando valores
```

```
   A: NEXT N
20 DATA 33,141,92,54,112,205
   ,107,13,62,2,205,1,22,1,0
   ,0,205,229,34,217,229,217
   ,58,171,195,14,0,71,17,1,
   1,205,186,36,58,171,195,
   79,6,0,17,1,1,205,186,36,
   58,171,195,71,14,0,17,1,
   255,205,186,36,58,171,195
   ,79,6,0,17,255,1,205,186,
   36,58,171,195,61,61,50,
   171,195,254,1,32,196,62,
   175,50,171,195,217,225,
   217,201,175
```

Para ejecutar la rutina RANDOMIZE USR 50000

MANEJO DE CIRCLE EN C/M

Igual que en el caso anterior, el manejo de esta rutina de la ROM requiere tener almacenados los valores necesarios en el lugar adecuado. La rutina CIRCLE empleada ($232D), opera con los 3 valores superiores contenidos en el *stack*, por lo que iremos introduciendo las coordenadas del centro y del radio mediante la rutina $2D28, que pasa el contenido A (previamente cargado con las coordenadas) al *stack*.

El efecto que se pretende conseguir es el equivalente al programa:

```
10 CLS: FOR N=0 TO 86 STEP 2
   : CIRCLE N,N,N: NEXT N
```

que dibuja una serie de circunferencias con el mismo valor para el radio y coordenadas «x» e «y» del centro.

(Ver en cuadro RUTINA 2.)

El cargador BASIC puede ser como este:

```
10 CLEAR 59999: FOR N=60000
   TO 60053: READ A: POKE N,
   A: NEXT N
20 DATA 205,197,13,62,2,205,
   1,22,217,229,217,58,149,
   234,205,40,45,58,149,234,
   205,40,45,58,149,234,205,
   40,45,205,45,35,58,149,
   234,60,60,50,149,234,254,
   88,32,223,62,0,50,149,234
   ,217,225,217,201,0
```

Para ejecutar la rutina RANDOMIZE USR 60000

RESERVA TU INPUT ESPECIAL VERANO

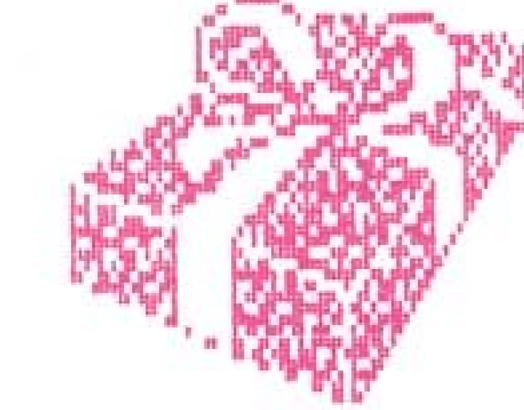
Que sorteará más de **100 REGALOS**

Reserva tu INPUT

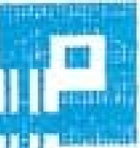


ESPECIAL de Verano





Sortearemos mas de 100 REGALOS SORPRESA



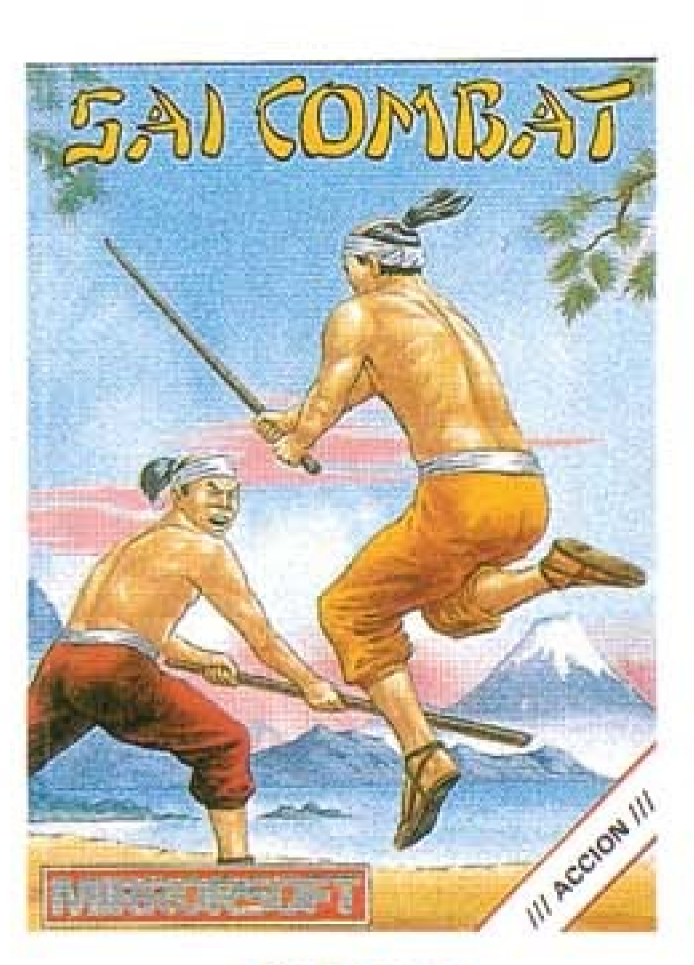
Ponte en Forma

Practica con los mejores programas deportivos

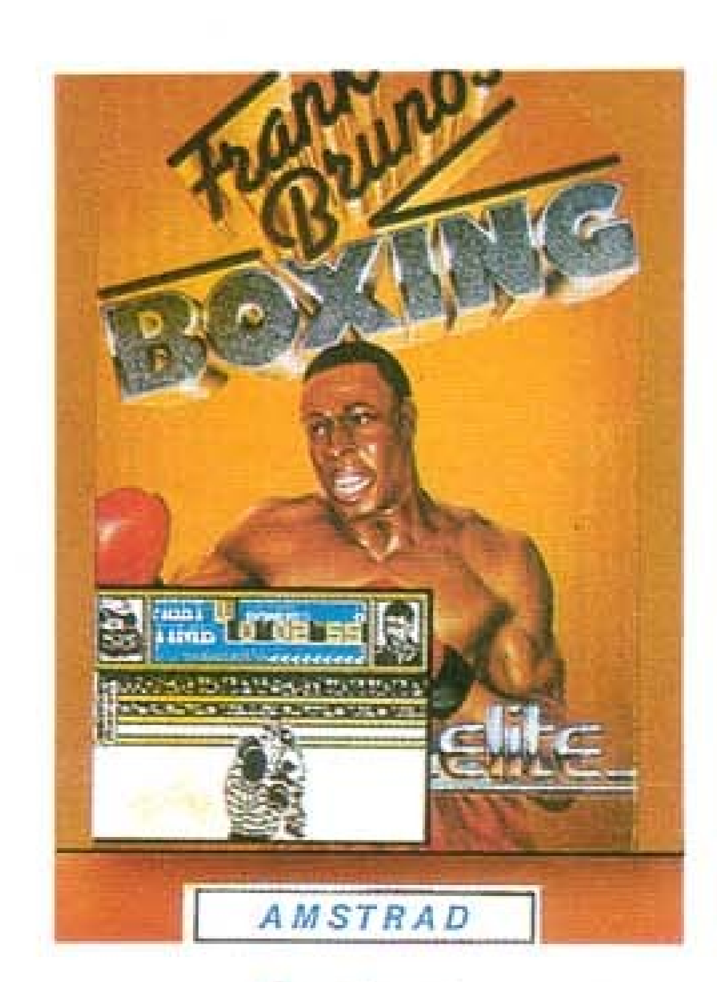


FUERZA
BOXEO
CICLISMO
PIRAGÜISMO
SQUASH

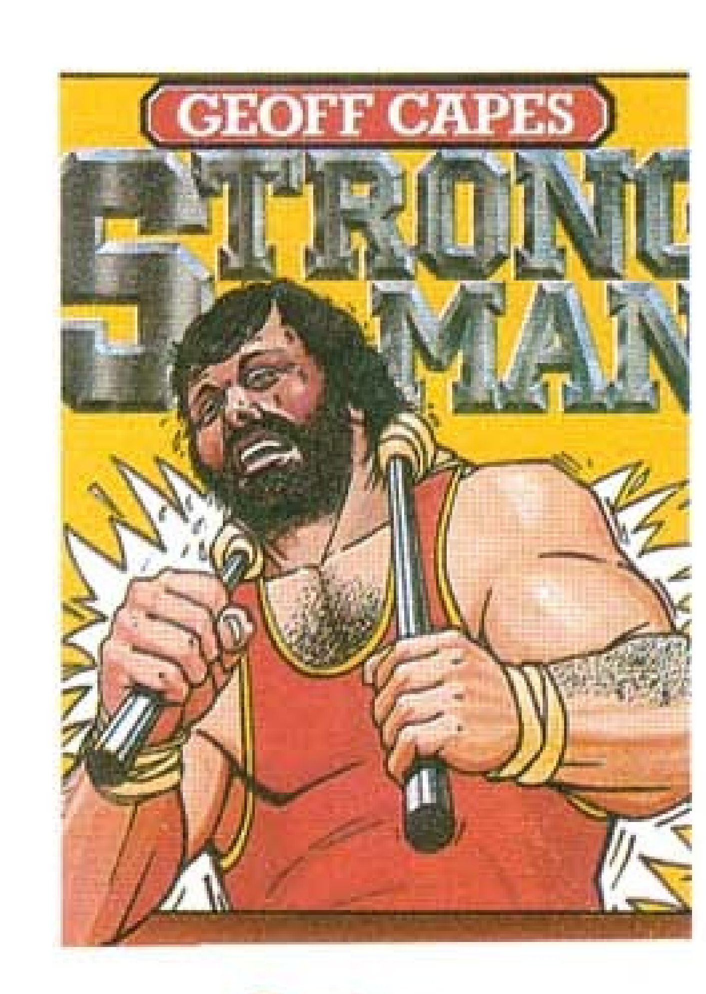
ARTES MARCIALES
BILLAR
GIMNASIA
HIPICA
TIRO



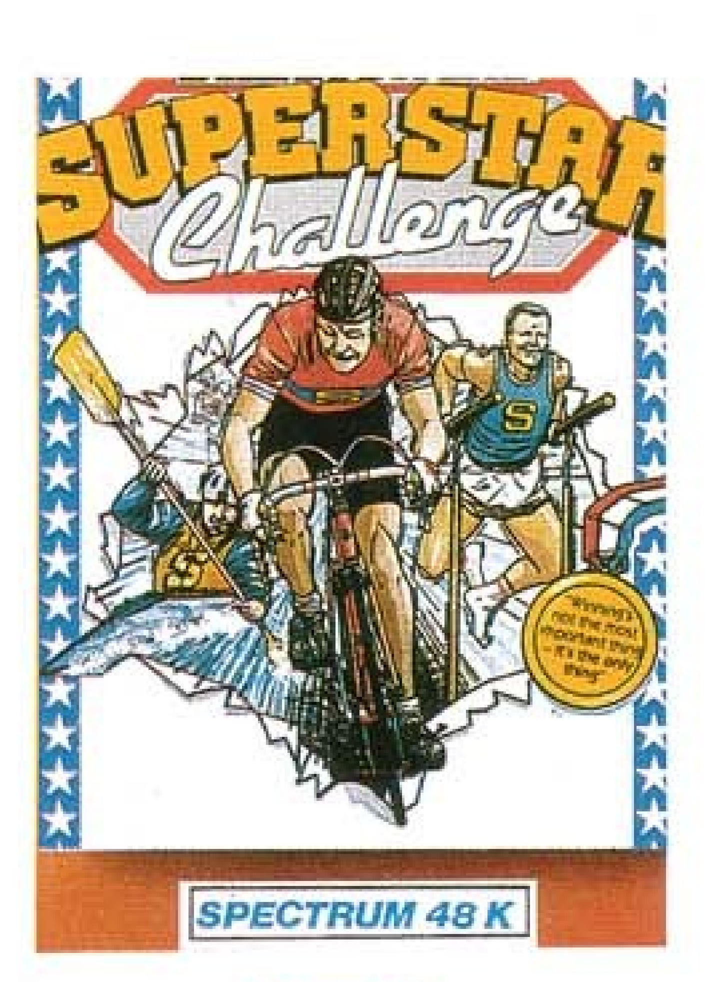
Spectrum



Spectrum
Commodore
Amstrad
Amstrad Disk



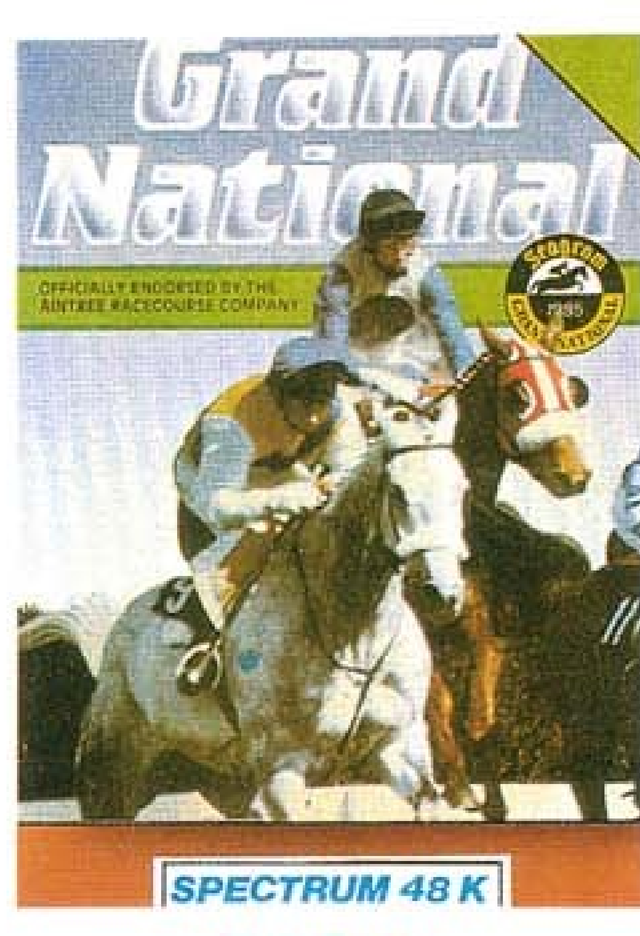
Spectrum
Commodore
Amstrad



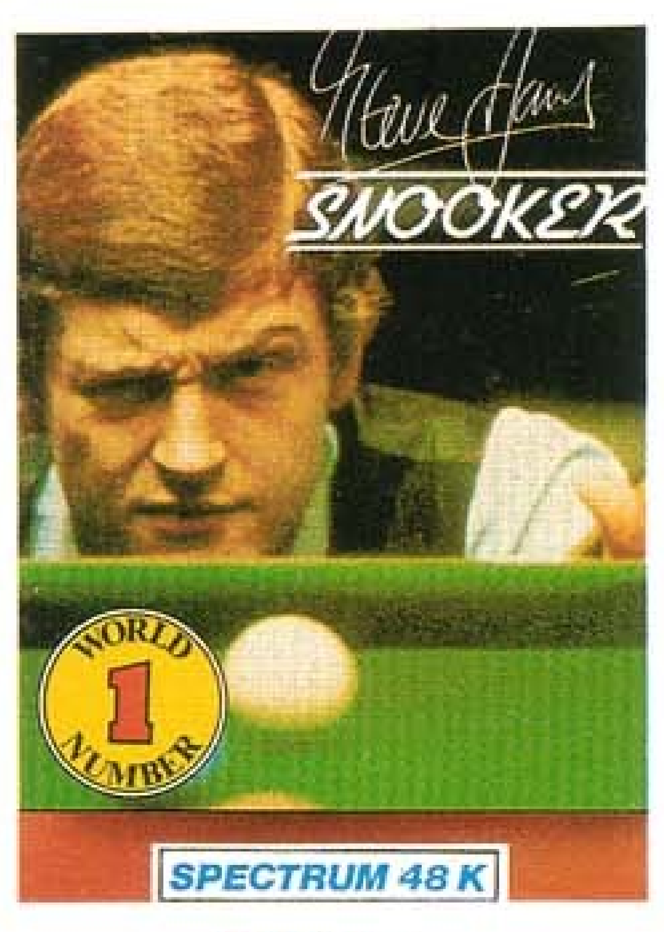
Spectrum
Commodore
Amstrad
MSX



Spectrum
Commodore



Spectrum



Spectrum
Commodore
Amstrad



ZAFIRO SOFTWARE DIVISION
Paseo de la Castellana, 141. 28046 Madrid.
Tel. 459 30 04. Tel. Barna. 209 33 65.
Telex: 22690 ZAFIR E

Programas editados, fabricados y distribuidos en España con la garantía Zafiro. Todos los derechos reservados.

COMO SE ALMACENAN LOS PROGRAMAS EN BASIC

La forma interna de trabajar el ordenador puede parecer a veces algo oscura, sobre todo cuando las cosas empiezan a no funcionar. Pero si consigues entender la manera en que se almacena un programa, estás en el buen camino para empezar a corregir sus errores.

¿Alguna vez te ha preocupado saber lo que hace el ordenador con un programa cada vez que lo tecleas? Como probablemente sabes, el ordenador lo almacena en una zona especial de la memoria, reservada para los programas en BASIC, pero puede que no sepas dónde está situada dicha zona, ni qué aspecto presenta el programa cuando se convierte a una forma que resulte comprensible por el ordenador.

De hecho el programa almacenado en la memoria del ordenador se parece mucho a los caracteres que tecleas y muchos de los números que contiene la memoria son los códigos ASCII de las letras del programa. Pero el ordenador acorta algunas palabras, las **palabras clave**, y además añade alguna información adicional, situada literalmente entre líneas, a fin de poder localizar posteriormente dónde está cada cosa.

El ordenador dispone de una zona separada en la que almacena las variables definidas por el programa y a medida que éstas cambian durante la ejecución las va manteniendo constantemente actualizadas. Así, cuando por ejemplo tecleas 10 LET A = 6, el ordenador almacena la línea como una parte de tu programa y además pone el nombre y el valor de la variable en el área de almacenamiento de variables.

Empecemos ya nuestra exploración. Primero tienes que introducir el siguiente programa. Tecléalo exactamente igual que te lo presentamos aquí. Cuida de no introducir espacios adicionales. Puedes ahora utilizar el programa para descubrir la manera en que el ordenador almacena sus programas.

```
10 PRINT "HOLA"
20 LET A=2
30 PRINT A
40 STOP
```

En el **Spectrum** hay dos direcciones de memoria que actúan conjuntamente como puntero del comienzo del programa en BASIC. Estas direcciones son las 23635 y 23636. Para encontrar el punto donde comienza tu programa BASIC, teclea el siguiente comando en modo directo:

```
PRINT PEEK 23635+256*PEEK
23636
```

Normalmente la respuesta será el número 23755 para un **Spectrum** de 48K. Si encuentras que tu ordenador te responde con un valor diferente, y el comando directo estaba correctamente tecleado, sustituye en el siguiente comando directo el número 23755 por el valor que hayas obtenido. Teclea el siguiente comando:

```
FOR I=23755 TO 23755+40:
PRINT PEEK I;" ";:NEXT I
```

Como resultado obtendrás la siguiente salida. De no ser así, comprueba si has tecleado correctamente:

```
0  10  9  0  245  34  72  79
76  65  34  13  0  20  11  0
241  65  61  50  14  0  0  2
0  0  13  0  30  3  0  245
65  13  0  40  2  0  226  13
```

Empleando ahora los códigos ASCII correspondientes a letras y números, mira a ver si le encuentras algún sentido a lo siguiente:

```
?  ?  ?  ?  ?  "  H  O  L  A
"  ?  ?  ?  ?  ?  ?  A  =  2
?  ?  ?  ?  ?  ?  ?  ?  ?  ?
?  A  ?  ?  ?  ?  ?  ?  A  ?
?  ?  ?  ?  ?  ?
```

Puedes ver cómo ya asoma una parte del programa en BASIC, aunque todavía tienes bastantes interrogaciones.

Tal vez te esperabas que delante de «HOLA» te apareciera la palabra PRINT, pero en su lugar la lista te muestra el valor 245. El **Spectrum** utiliza los valores superiores a 164 para las **palabras clave** del BASIC. Si consultas en tu manual la lista correspondiente, encontrarás que 245 es el valor (el llamado *token*) que corresponde a PRINT. Examinando los valores de los *tokens* puedes localizar las **palabras clave**:

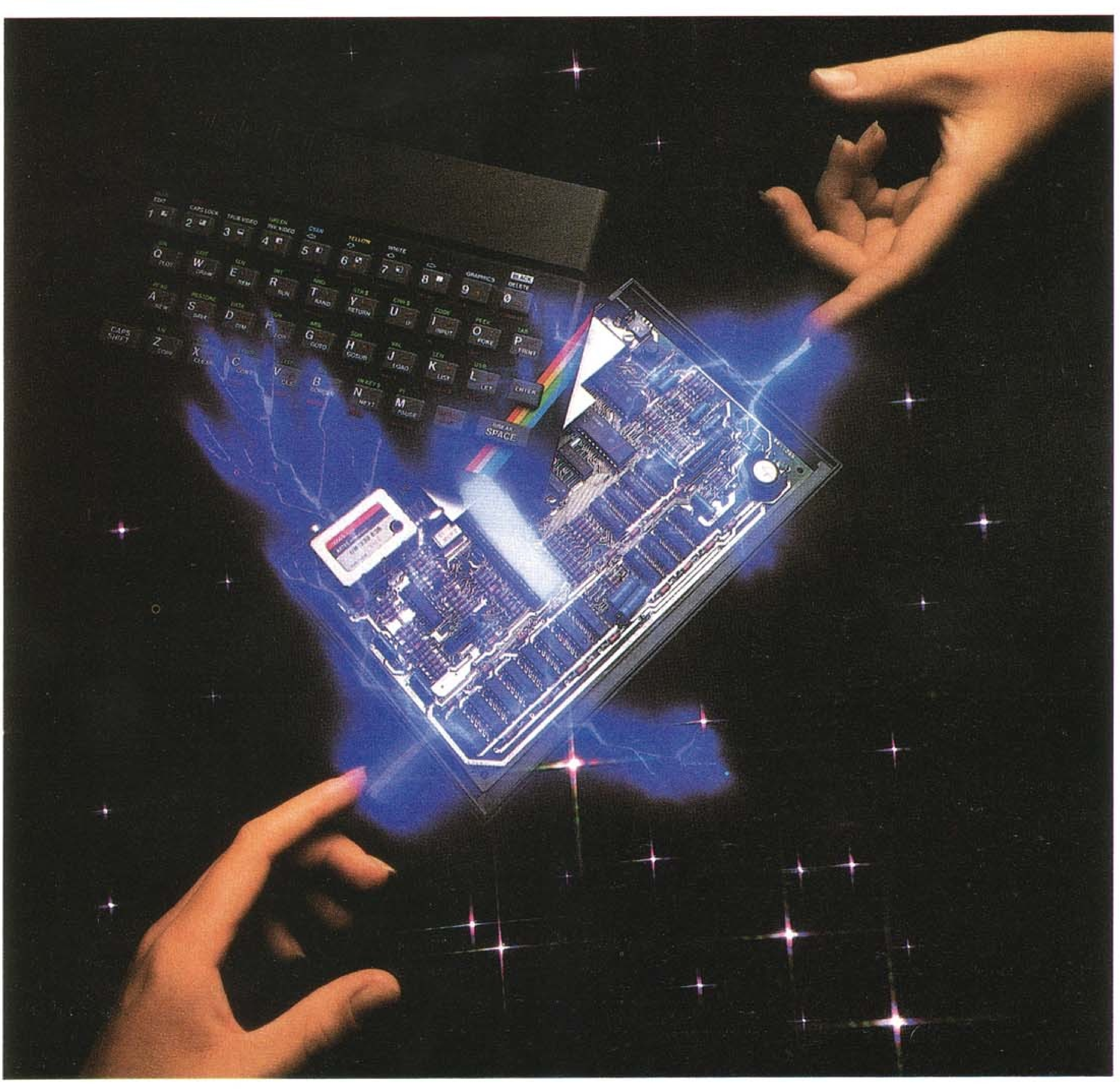
```
?  ?  ?  ?  PRINT"  H  O  L
A  "  ?  ?  ?  ?  ?  LET  A
=  2  ?  ?  ?  ?  ?  ?  ?
?  ?  ?  ?  PRINT  A  ?  ?
?  ?  ?  STOP  ?
```

Si te fijas ahora en los bytes 2°, 14°, 29° y 36° de la tabla original, detectarás la secuencia 10, 20, 30, 40. Estos bytes se utilizan para almacenar la parte menos significativa de los números de línea, mientras que los bytes que les preceden contienen la parte más significativa. Los dos bytes siguientes de la línea, esta vez con el menos significativo en primer lugar, se utilizan para almacenar la longitud de la línea, (longitud byte Bajo, longitud byte Alto) excluyendo estos primeros cuatro bytes.

Ahora dispones de la siguiente imagen:

- LOCALIZACION DEL INICIO DEL BASIC
- CODIGOS ASCII Y TOKENS
- COMO SE ALMACENA EL PROGRAMA
- VARIABLES NUMERICAS

Vayamos ya con los bytes finales. El valor 13 se utiliza para indicar el final de línea (fdl). El 14 indica que los cuatro bytes siguientes son la forma codificada que corresponde a una constante numérica, que en este caso es un 2. El código 14 se utiliza para indicar una



```
0  10  lonB  lonA  PRINT"  H
0 L  A  "  ?  0  20  lonB
lonA  LET  A  =  2  ?  ?  ?
?  ?  ?  ?  0  30  lonB  lonA
PRINT A  ?  0  40  lonB  lonA
STOP  ?
```

constante numérica en la tabla que contiene la lista de variables (tlv). Aquí tienes pues el resultado final:

```
0  10  lonB  lonA  PRINT"  H
0  L  A  "  fdl  0  20  lonB
lonA  LET  A  =  2  tlv  0
0  2  0  0  fdl  0  30  lonB
lonA  PRINT  A  fdl  0  40
lonB  lonA  STOP  fdl
```

Para señalar el final de un programa, el **Spectrum** almacena un número mayor de 40 en el byte alto del campo que contiene el número de línea.

El comando NEW borra cualquier programa antiguo que pueda haber en el ordenador, así como todas las variables. Después de ejecutarlo ya no

las puedes recuperar, de modo que ten cuidado al utilizarlo.

ALMACEMAMIENTO DE LAS VARIABLES

El programa anterior muestra la forma en que el programa BASIC estaba almacenado en la memoria; sin embargo todas las variables definidas en el programa se almacenan separadamente en una zona especial para el almacenamiento de variables. Hay dos direcciones de memoria, las 23627 y 23628, que contienen la dirección donde comienza esta zona de almacenamiento, y otras dos direcciones —23641 y 23642— que contienen la dirección donde acaba dicha zona.

Ensaya el siguiente programa que sirve para imprimir todo el contenido de dicha zona de almacenamiento:

```
10 FOR A=PEEK 23627+256*PEEK
23628 TO PEEK 23641+256*PEEK
23642
20 PRINT PEEK A;"  ";
30 NEXT A
```

VARIABLES NUMERICAS

En primer lugar define una variable:

```
1 LET B=150000
```

Al ejecutar el programa, obtienes la siguiente salida:

```
98 146 18 124 0 0 225 0
 B .....valor....
```

El 98 es el código ASCII de B más 32; esto le indica al ordenador que está manejando una variable numérica. El 150000 se almacena en los cinco bytes siguientes con formato de punto flotante. Los dos últimos bytes, 225 y 0, señalan el final de la zona de almacenamiento de variables. Siempre figuran de la misma manera, por lo que los omitiremos en los siguientes ejemplos.

Los nombres largos de variables se almacenan de la misma forma, aunque con alguna complicación suplementaria. Ensaya lo siguiente:

```
1 LET DAVID=30
```

Esto da el siguiente resultado:

```
164 97 118 105 228 0 0 30 0 0
 D   A   V   I   D ...valor..
```

Los cinco primeros bytes sirven para almacenar el nombre DAVID, con la salvedad de que al código ASCII de la primera letra se le ha sumado 96 para indicar que se trata de un nombre largo, a los centrales se les ha sumado el 32 habitual y al último se le han sumado 160 para indicar el final del nombre. En los últimos cinco bytes se almacena el número.

MATRICES Y CADENAS DE CARACTERES

La siguiente línea de programa demuestra la forma en que se almacenan las cadenas de caracteres:

```
1 LET A$="CADENA"
65 6 0 67 65 68 69 78 65
 A      C  A  D  E  N  A
```

En primer lugar figura el nombre de la cadena de caracteres, A, seguido de dos bytes que dan la longitud del texto. Inmediatamente después siguen los códigos ASCII de la propia cadena.

Veamos ahora lo que ocurre con una matriz numérica:

```
1 DIM F(2)
2 LET F(1)=100
3 LET F(2)=200
134  13  0  1  2  0  0  0
 F                     /...
100  0  0  0  0  200  0  0
valor.../  /....valor..../
```

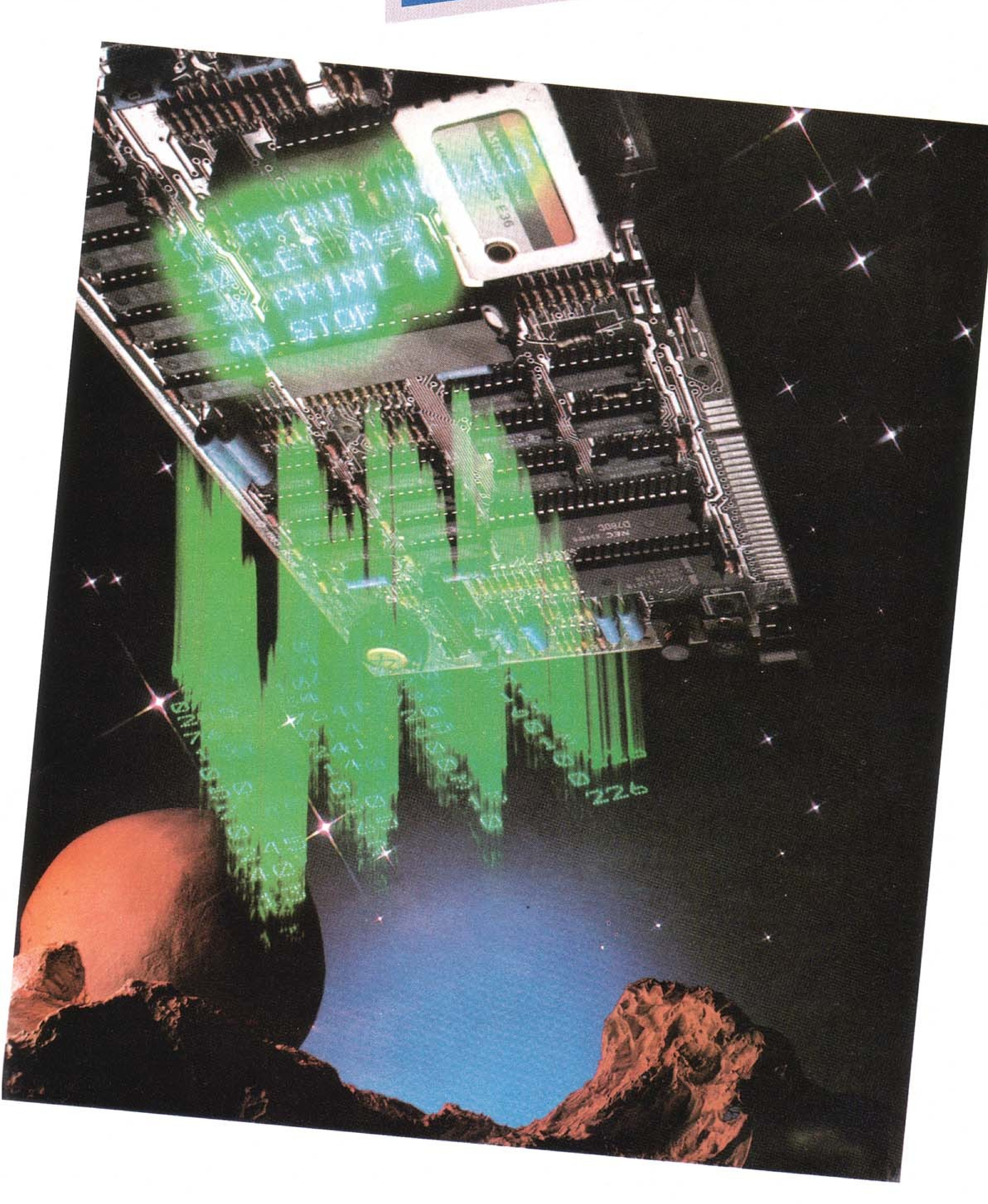
El primer byte es el código de F más 64, que corresponde a una matriz numérica. Los dos siguientes contienen el número de bytes que vienen a continuación, empezando por el byte bajo. Cuéntalos y verás que son 13. El siguiente byte da el número de dimensiones y los dos siguientes almacenan el número de elementos reservados. A continuación vienen dos números, cada uno de ellos almacenado en cinco bytes.

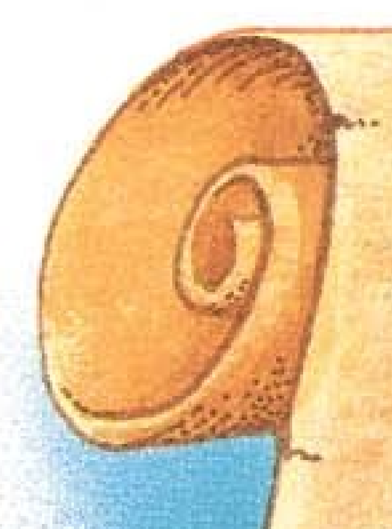
Por último, prueba con una matriz de cadena de caracteres:

```
1 DIM B$(2,6)
2 LET B$(1)="CADENA"
3 LET B$(2)="MATRIZ"
194  17  0  2  2  0  6  0
 B
67  65  68  69  78  65  77
 C   A   D   E   N   A   M
65  84  82  73  90
 A   T   R   I   Z
```

El primer byte es el código ASCII de B incrementado en 128, que corresponde a las matrices de cadena de caracteres. Los dos siguientes dan el número de bytes que vienen detrás. El siguiente byte contiene el número de dimensiones, que en este caso es 2, seguido de dos pares que dan el número de elementos de la matriz y la longitud máxima de cada elemento. A continuación siguen inmediatamente las propias cadenas. El espacio que figura al final se añade a la palabra MATRIZ para terminar de llenar los seis caracteres asignados a cada cadena.

Si se te hace difícil encontrar INPUT en tu kiosco habitual, resérvalo por adelantado, o háznoslo saber para que podamos remediarlo


PRINT A
STOP
226



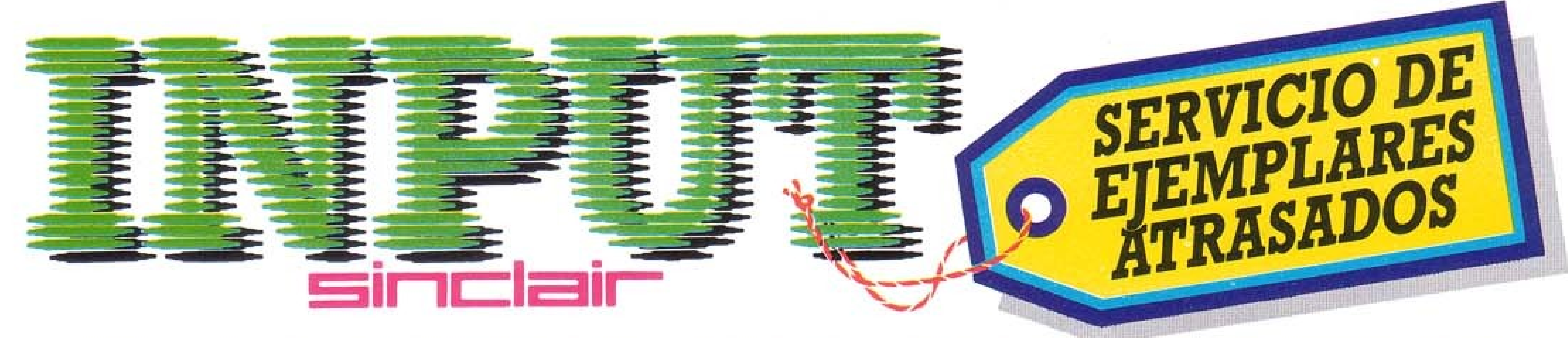
LOS MEJORES DE INPUT SINCLAIR

PUESTO	TITULO	PORCENTAJE
1.°	Commando	20,6 %
2.°	Rambo	13,6 %
3.°	The Dambusters	11,3 %
4.°	Camelot Warriors	10,4 %
5.°	Skyfox	10,2 %
6.°	Movie	9,1 %
7.°	Sir Fred	7,9 %
8.°	Saboteur	6,8 %
9.°	Turbo Esprit	5,6 %
10.°	Amazon Woman	4,5 %
		100 %

Para la confección de esta relación únicamente se han tenido en cuenta las votaciones enviadas por nuestros lectores de acuerdo con la sección «Los Mejores de Input».

Julio de 1986





¡NO TE PIERDAS NI UN SOLO EJEMPLAR!

INPUT SINCLAIR quiere proporcionar a sus lectores este nuevo servicio de ejemplares atrasados para que no pierdan la oportunidad de tener en sus hogares todos los ejemplares de esta revista, líder en el mercado español.

Podréis solicitar cualquier número de **INPUT SINCLAIR** que querais, siempre al precio de cubierta (sin más gastos).

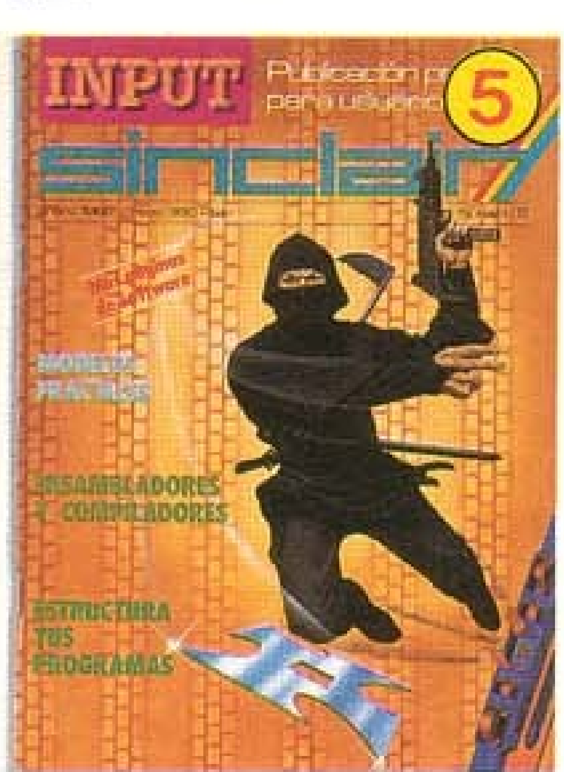
Utiliza el cupón adjunto, enviándolo a **EDISA** (Dpto. de Suscripciones), López de Hoyos, 141 - 28002 Madrid, o bien llámanos por teléfono al (91) 415 97 12.























CUPON DE PEDIDO

SI, envíenme contrarreembolso .... ejemplares de **INPUT SINCLAIR** de los números:

(marca con una (X) tu elección)

☐ 1 ☐ 2 ☐ 3 ☐ 4 ☐ 5 ☐ 6 ☐ 7 ☐ 8 ☐ 9 ☐ 10

NOMBRE

APELLIDOS

DOMICILIO

NUM. PISO ESCALERA COD. POSTAL

POBLACION PROV.

TELEFONO FIRMA

EL GUERRERO INVENCIBLE

Estamos en el siglo XXV, cuando los vasallos del planeta Kindos han robado todo el oro que encerraba el Palacio Real y han raptado a la princesa **Doyaphin**.

El objetivo es rescatar a la princesa que se encuentra en el castillo de Kindos, para lo cual el guerrero **Mirdav** debe introducirse en la fortaleza y salvar todos los obstáculos que se le presenten. Tras bajar una pequeña rampa comienza la aventura, apareciendo multitud de seres dispuestos a estropear la misión.

Los enemigos con los que se va a encontrar son grandes arañas, ratones asesinos, serpientes venenosas, esqueletos en movimiento, etc. La mayoría de éstos aparecen por la parte inferior de la pantalla por lo que con un simple salto se pueden evitar. Los problemas comienzan cuando surge el mosquito volador ya que hay que agacharse en el momento preciso, de lo contrario es muy fácil perecer.

Para sobrepasar a los guardianes del castillo contamos con **Sgrizam**, una espada mágica que permite vencer a cualquier guerrero que se presente.

A medida que se introduce en el castillo, va superando distintas fases, cada vez más complicadas, hasta llegar al calabozo donde se encuentra encerrada la princesa.

Esta se encuentra custodiada por un dragón que con un simple golpe de espada, puede ser eliminado. Pero en este momento se ha cumplido la mitad de la misión, a continuación hay que salir del peligroso castillo.

El guerrero **Mirdav** puede saltar, agacharse, avanzar y atacar con su poderosa espada, figurando en pantalla el número de vidas y el tiempo que ha transcurrido.

Dinamic ha construido un clásico juego de habilidad en donde es necesario contar con una gran sangre fría ya que hay que pulsar la tecla en el momento apropiado.


DATOS GENERALES

TITULO Sgrizam

FABRICANTE Dinamic

ORDENADOR Spectrum 48K

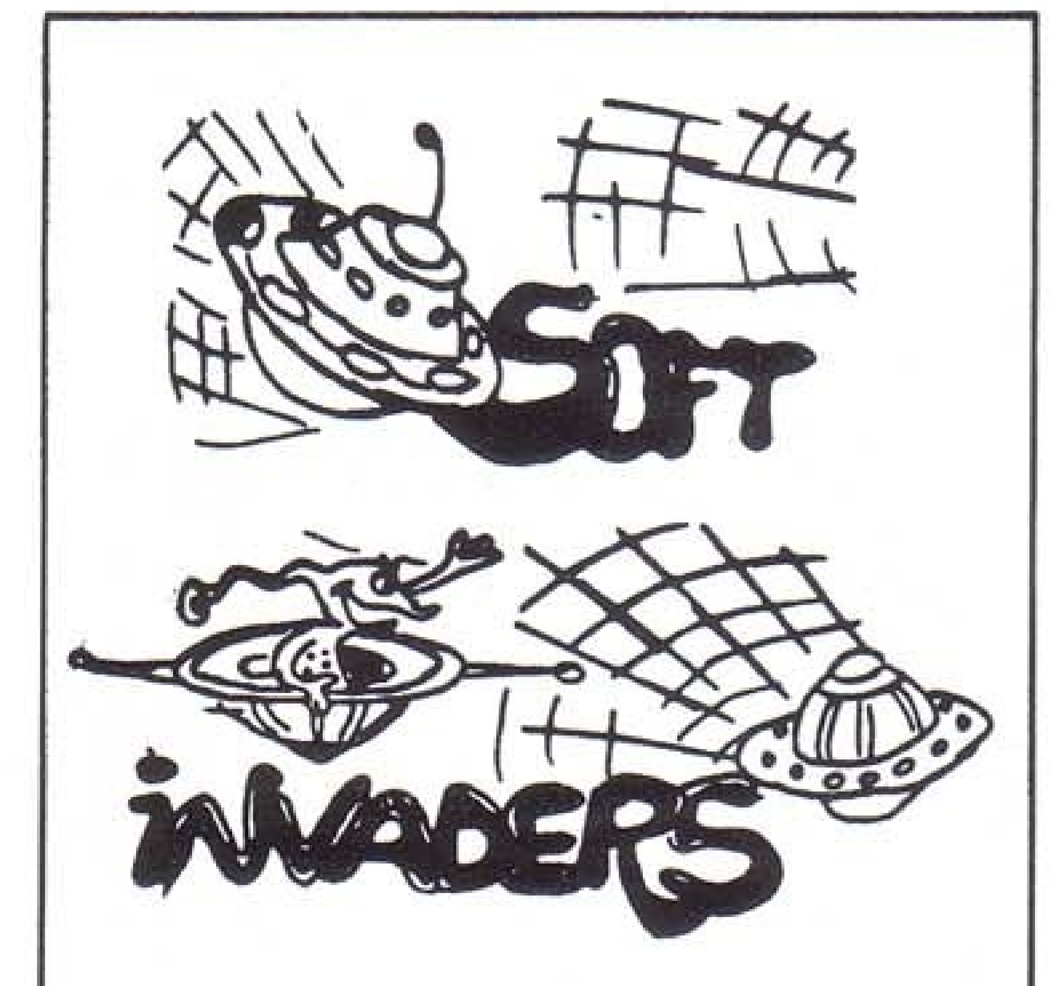
TEMA DEL PROGRAMA
Rescate peligroso

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	6
GRAFICOS	8
COLOR	8
SONIDO	7
TOTAL	35







SPECTRUM

Nombre	P.VPC
ALIEN 8	2190
BEACH HEAD II	1990
BOMBJACK	1950
COMBAT LYNX	1860
COMET (HOT)	1673
COMMANDO	1700
CRASH SMASHES	1930
CRITICAL MASS	1850
ENIGMA FORCE	2272
EQUINOX	1930
FAIRLIGHT	1900
GREEN BERET	1600
GYROSCOPE	1900
HEAVY ON THE MAGIC	1930
HOCUS FOCUS	1799
KNIGHT LORE	2190
MANTRONIX	1669
MATCH DAY	2025
MISION IMPOSIBLE	1850
NIGHT SHADE	2190
NOW GAMES II	2135
QUAZATRON	1799
RAMBO	1700
ROBIN OF SHERWOOD	1700
SABOTEUR	1850
SAMANTHA FOX	1799
SKYFOX	2250
STARQUAKE	1673
SWORDS & SORCERY	2437
TAU CETI	1930
THEY SOLD A MILLION	2135
THINK	2217
TOMAHAWK	1930
TURBO ESPRIT	1850
V	1673
VIERNES 13	1850
WEST BANKS	1978
WINTER GAMES	1970
YIE AR KUNG FU	1750
ZORRO	1800

ESTOS PRECIOS NO INCLUYEN EL IVA

NUESTRAS OFERTAS SON NUESTROS PRECIOS

ENVIANOS A
SOFT INVADERS ORENSE 6 28020 MADRID
DESEO RECIBIR CONTRA-REEMBOLSO LOS SIGUIENTES JUEGOS:

NOMBRE	CANT.	PRECIO
	TOTAL + 175 (gastos envío)	

ME LOS ENVIAN A:
NOMBRE:
DIRECCION:
POBLACION: COD. POSTAL:
TELEFONO:
FIRMA

CONTRA EL IMPERIO DE LA DROGA

DATOS GENERALES

TITULO Turbo Esprit

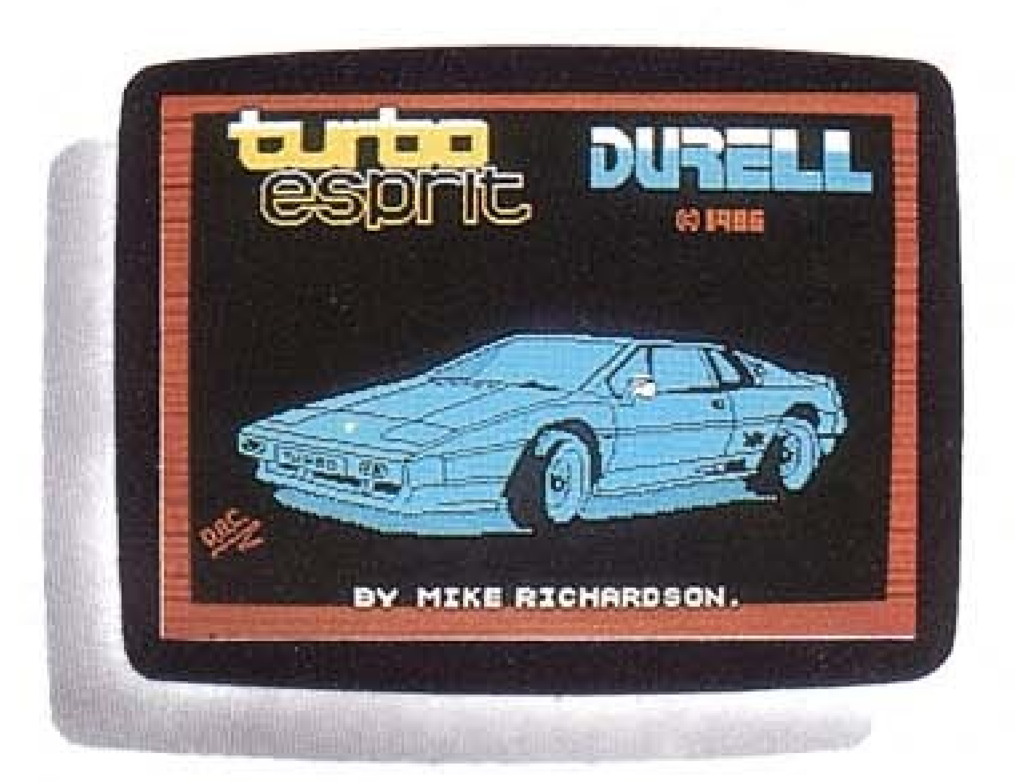
FABRICANTE Durell

ORDENADOR Spectrum 48 K

TEMA DEL PROGRAMA
Persecución policial

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	**8 sobre 10**
INTERES	**8 sobre 10**
GRAFICOS	**7 sobre 10**
COLOR	**5 sobre 10**
SONIDO	**3 sobre 10**
TOTAL	31





En cuatro ciudades norteamericanas se produce cada día un organizado tráfico de estupefacientes. Un vehículo blindado lleva en su interior un alijo de drogas, cuyos destinatarios son cuatro coches de reparto, que una vez realizada la entrega huirán hacia sus escondrijos. Tras el intercambio se inicia la búsqueda de los coches por toda la ciudad para intentar interceptarlos e impedir que el cargamento llegue a su destino.

Los coches tienen un color característico que permite su identificación, pese a lo cual localizarlos en una gran ciudad sería muy difícil. Sin embargo, la policía cuenta con agentes instalados en lugares estratégicos que informan al paso de los delicuentes, por lo que una vez conocida su situación hay que dirigirse rápidamente hacia el sector en que han sido vistos.

En pantalla puede observarse la dirección, velocidad, revoluciones del motor, indicadores de gasolina y presión, y una vista panorámica de la calle como si estuvieramos en el interior del coche.

Para un mejor conocimiento de la ciudad se dispone de un mapa que facilita información sobre las gasolineras más próximas y orienta sobre la distribución de las calles.

Al comenzar el juego selecciona una de las cuatro ciudades posibles, el nivel de dificultad y las teclas o *joysticks* a utilizar, disponiendo de la posibilidad de practicar el control del automóvil.

Un potente **Turbo Esprit,** capaz de alcanzar las 180 millas por hora, perseguirá implacable a los malhechores hasta darles alcance.

Pero conducir por una ciudad llena de vehículos y peatones resulta difícil, además, los semáforos, las intersecciones, los tramos en obra, los pasos cebra y las calles sin salida dan un mayor realce y peligrosidad a la acción.

El automóvil puede ir hacia adelante y hacia atrás, girar a izquierda y derecha, frenar y acelerar, e incluso disparar para detener a los traficantes.

La entrada en acción hay que posponerla hasta que se halla consumado la entrega y para la rendición de los «camellos» basta con situarse tras ellos y golpear tres veces su coche o bien dispararles, aunque esto último es conveniente evitarlo.



Este juego presentado por **Durell** encierra un gran realismo y además de las anteriores características hay que mencionar la posibilidad de grabar (y posteriormente cargar) la tabla de *récords.* Eso sí, se trata de una conducción a la inglesa, es decir por la izquierda.

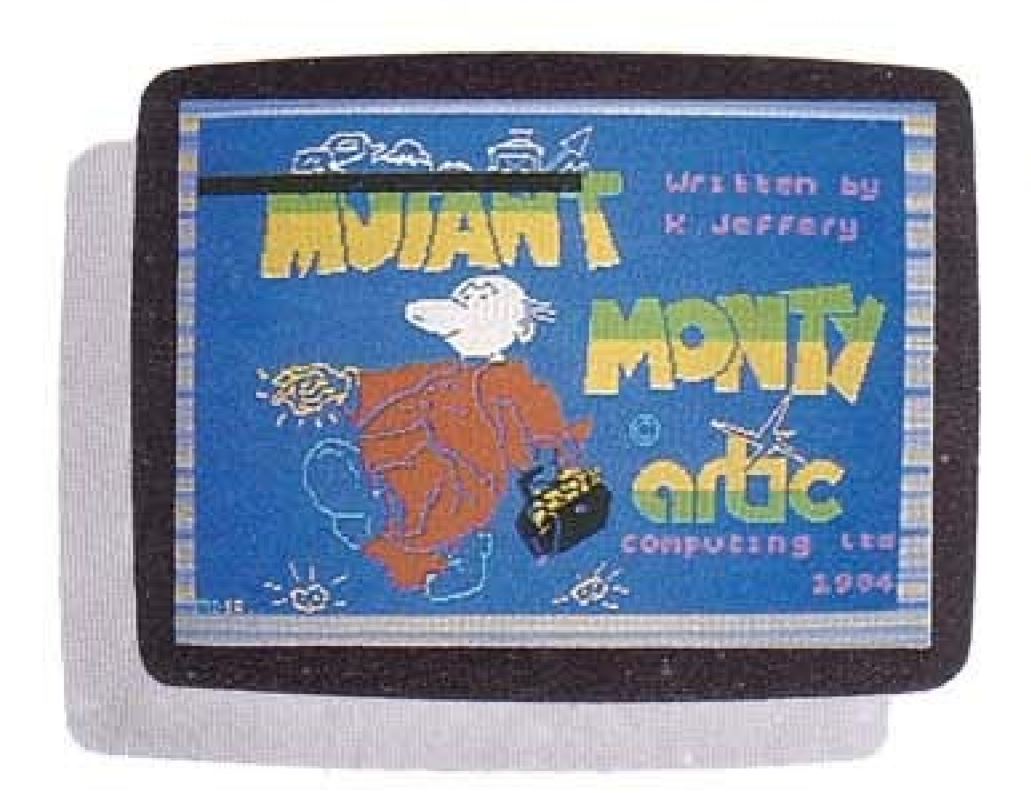
****

GRANDES EXITOS

No es habitual reunir en una sola oferta diez juegos diferentes, pero si han alcanzado un cierto renombre en las listas de éxitos, es indudable que cuente con un gran atractivo. Los diez programas se presentan en dos *cassettes* con cinco juegos cada una, que van desde complicados laberintos tridimensionales, hasta el clásico conversacional, pasando por los inevitables juegos espaciales.

Codename Mat es uno de los programas más complejos existentes en el mercado y para confirmarlo basta decir que pueden utilizarse veinte teclas en cada juego. Se trata de una misión espacial que busca la destrucción de los alienígenas, una







creación de **Micromega** que requiere de algún tiempo para hacerse con el control de todas las posibilidades que encierra.

Wizard's Lair es un entretenido juego de **Bubblebus** que tiene a un fantasma como protagonista, cuya misión es encontrar las cuatro piezas del León Dorado. Esta es la única posibilidad para salir del laberinto de cuevas, habitaciones, túneles, pasajes, puertas, ríos subterráneos, lagos, cataratas y extraños seres que habitan en este lugar. Cuenta con unos atractivos gráficos llenos de color y nos recuerda al conocido **Atic Atac.**

Un joven experto en ordenadores es el protagonista de **Technician Ted,** un juego de **Hewson** donde el personaje debe completar sus 21 tareas diarias antes de las cinco de la tarde. Se trata de una fábrica de *chips* en la que según las tareas realizadas y el tiempo empleado se obtendrá una mayor o menor puntuación.

Mutant Monty tiene que atravesar cuarenta habitaciones hasta conseguir su objetivo, recolectando oro y rescatando a damiselas en peligro. Todo ello en un tiempo limitado y con sólo cinco vidas; la firma **Artic** no lo ha puesto nada fácil.

Snooker es un original juego de salón que presenta una mesa de billar americano en la que tiene lugar una partida con uno o dos jugadores. Se puede seleccionar el número de bolas, la dirección y la intensidad del golpe. Un excelente juego de **Visions** que con muy poca memoria consigue un gran realismo.

Un laberinto tridimensional es el escenario de **Androide 2,** una creación de **Vortex** donde la rapidez y la habilidad son fundamentales. La misión es detener a los **Millitoids** y sobrevivir en este mortal laberinto.

El argumento de **On the Run,** un programa creado por **Design Desinyng,** es conseguir seis frascos de un peligroso producto químico que tiene la propiedad de realizar mutaciones en animales y plantas. Por ello, antes de que esto ocurra hay que localizar, en una zona llena de peligros, esos productos letales.

Los restos de una antigua civilización se encuentran dispersos por un complejo subterráneo, por lo que para conocer ese pueblo perdido en el tiempo hay que ensamblar todos los fragmentos. **The Covenat** es un juego de **PSS** que tiene a un astronauta como protagonista y que cuenta con la posibilidad de grabar la posición en que se encuentra para continuar el juego en cualquier otro momento.

En **Super Pipeline 2** lo único que se debe hacer es vigilar la tubería por la que pasa el agua que llena unos barriles. Esto que parece sencillo, se complica con la aparición de peligrosos personajes que dificultan el objetivo de este juego creado por Tasket.

Por último el programa **Circus** es un juego conversacional para el que es necesario tener algunos conocimientos de inglés. Se trata de un coche sin gasolina, en una solitaria carretera, teniendo que buscar a alguien que le ayude; en esta situación observa un circo ambulante hacia el que se dirige.

Beau-Jolly ha conseguido diez interesantes juegos que todavía cuentan con un indudable interés.

****

ENCUENTRA EL BATICRAFT

Mantener la paz en el mundo es el objetivo de dos conocidos personajes: **Batman** y **Robin**. Pero siempre hay alguien dispuesto a poner dificultades, y para ello han secuestrado a **Robin**, al que tienen oculto en las bati-cuevas, debajo de **Gotham City**. Para rescatarle es necesario utilizar el **Baticraft** que se encuentra desmontado en siete piezas esparcidas por todas las habitaciones, siendo el objetivo del juego su localización. Pero antes de montar el **Baticraft** hay que conseguir las botas para saltar, el bolso para coger y llevar artículos, el motor para el control horizontal y el cinturón que reduce la velocidad de las caidas. Con lo cuatros objetos en su poder **Batman** puede iniciar la búsqueda de las siete piezas del **Baticraft** que una vez ensambladas permitirán la huida de las cuevas. La aventura se inicia con nueve vidas, y a lo largo de las 150 habitaciones repartidas en nueve pisos, hay un total de 22 vidas extras que es

posible conseguir, así como unos poderes extraordinarios que no tienen mucha duración.
Una característica importante que ofrece el juego son las señales, que tras ser tocadas graban la situacion actual de la partida, permitiendo volver a ella en el caso de perder todas las vidas.
En el margen izquierdo de la pantalla aparecen los objetos



conseguidos, el numero de vidas que restan, el contador de saltos dobles y los indicadores de invulnerabilidad y supervelocidad. En el lado derecho aparecen los cuatro objetos iniciales que es necesario conseguir, cambiando de color a medida que son obtenidos.
Se puede con una misma tecla, saltar y coger los objetos, pese a que no es posible transportalos y es necesario

DATOS GENERALES

TITULO Batman

FABRICANTE Ocean

ORDENADOR Spectrum 48K

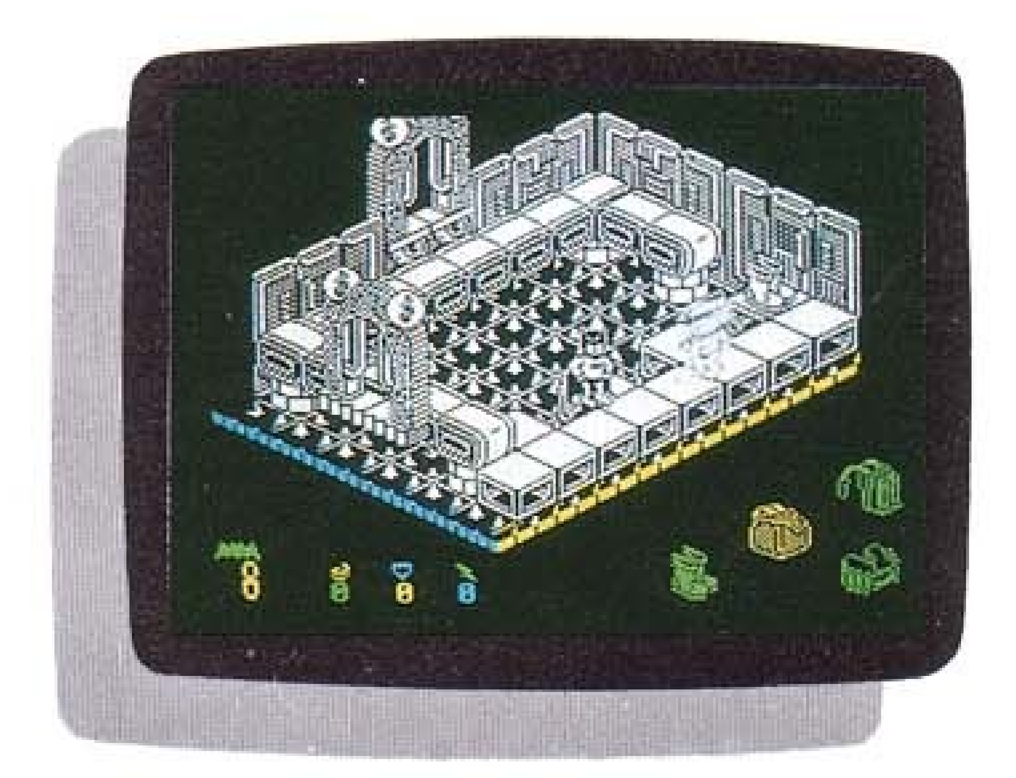
TEMA DEL PROGRAMA
Laberinto tridimensional

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	7
INTERES	7
GRAFICOS	8
COLOR	5
SONIDO	8
TOTAL	35



utilizarlos en la misma habitación.
Como en la mayoría de los juegos, cada vez que se traspasa una puerta, los objetos y personas que la habitan vuelven a su disposición original.
Ya en el menú se ofrecen originales alternativas como ajustar el volumen de sonido segun tres niveles y el control de sensibilidad para una



mayor respuesta del teclado o *joystick*.
Los decorados están muy bien conseguidos y los malvados que dificultan la aventura son unos peronajes francamente desagradables.
El programa ha sido creado por **Ocean** y tiene los ingredientes necesarios para aparecer en las listas de éxitos.

****

LA FORTALEZA ESTELAR

El poderoso **Goldter** ha conseguido amasar una enorme fortuna que tiene celosamente guardada en una inexpugnable mansión.
Phantomas, que logró escapar del control cibernético al que estaba sometido, es un ladrón profesional en busca de dinero y conoce los tesoros escondidos tras los muros.
El objetivo del juego es accionar un total de 36 palancas repartidas por la mansión, tras lo cual sonará una alarma; pero además hay que apoderarse de unas joyas guardadas en la cámara acorazada que inmediatamente se cerrará siendo muy difícil salir de allí. La clave para conseguir las joyas y salir de la cámara es solucionar el acertijo que figura en el manual.
La fortaleza cuenta con una base espacial, el palacio y el subterráneo que son los tres submundos que componen el laberinto creado por el malvado **Goldter**.
La base espacial está constituida en la parte más alta del edificio y en ella podemos encontrar helicópteros, antenas de radar o naves espaciales.
Si montamos en el helicóptero nos traladará a otro escenario, donde tras accionar una palanca hay que «poner los pies en polvorosa» ya que





una enorme piedra puede aplastar al protagonista. El cohete nos transporta a un paisaje lunar, con la tierra al fondo, donde hay que alcanzar la palanca y retornar a la nave.
Una vez dentro del palacio hay que atravesar un sin fin de habitaciones e ir accionando todas las palancas que surjan, intentando, no ser rozado por los seres y objetos que están en constante movimiento.
En el mundo subterráneo habitan innumerables seres muy peligrosos, cuyo simple contacto provoca la disminución de energía, además esta zona es más complicada que las



DATOS GENERALES

TITULO Phantomas

FABRICANTE Dinamic

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Búsqueda del Tesoro

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	7
GRAFICOS	7
COLOR	7
SONIDO	9
TOTAL	36

anteriores, ofreciendo muchas dificultades para moverse a lo largo de ella.
Cuando **Phantomas** es alcanzado por



algunos de los objetos o seres móviles, sus energías disminuyen y si son consumidas acaba el juego.
Sin embargo es posible recuperar fuerzas, ya que a lo largo del recorrido hay instaladas varias zonas de aprovisionamiento.
El personaje puede moverse de izquierda a derecha, así como saltar muy alto o muy largo, apareciendo en pantalla el termómetro de sus energías.
El programa presentado por **Dinamic** incluye una atractiva música y un buen color, pudiéndose utilizar varios *joysticks*.

★★★★★★★★★★★★★★★★★★★★★★★★★★★★

FIN DE CURSO

DATOS GENERALES

TITULO Tommy

FABRICANTE Future Stars

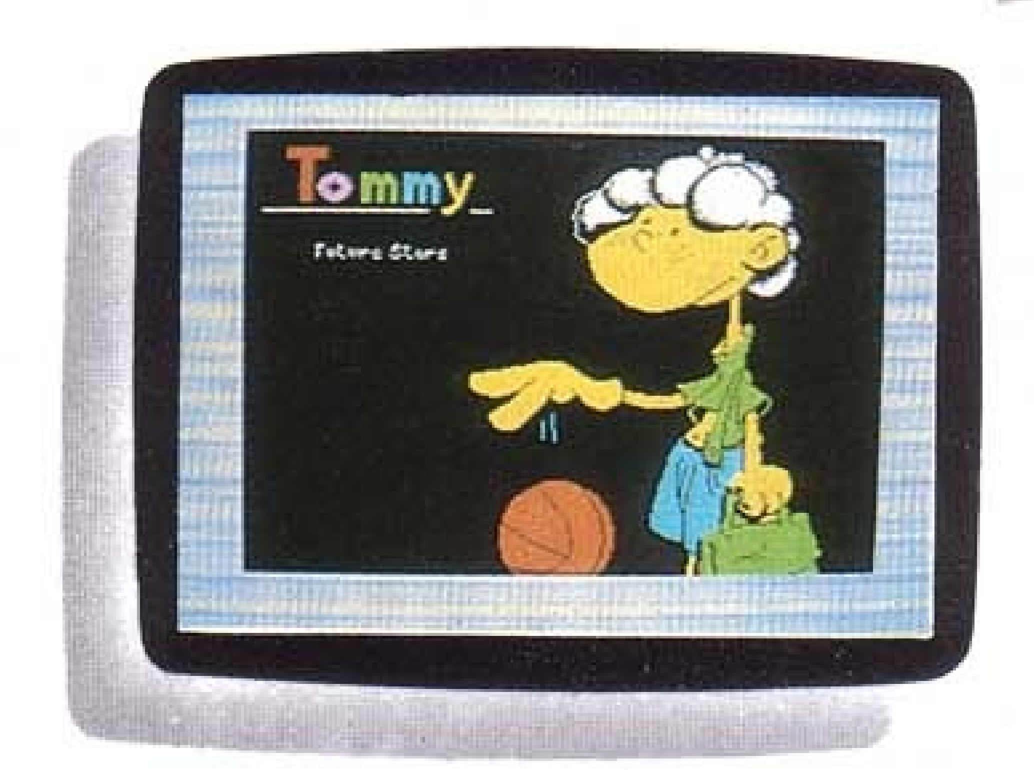
ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Aventura Colegial

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	6
INTERES	7
GRAFICOS	8
COLOR	8
SONIDO	8
TOTAL	37

El bueno de **Tommy** está preparando su viaje fin de curso, pero antes de ello debe localizar las notas obtenidas y depositarlas en el aula de viaje.
Cinco son las asignaturas que debe localizar: Literatura, Matemáticas, Ciencias, Química y Dibujo, cada una de las cuales precisa de una estrategia determinada.
La aventura se desarrolla en un colegio al que no le falta de nada, contando con biblioteca, gimnasio, sala de ordenadores, bar, secretaría, servicios... y por supuesto las aulas de Literatura, Matemáticas, Ciencias, Dibujo, Laboratorio y la correspondiente al viaje fin de curso.
A lo largo del recorrido van apareciendo diversos objetos, algunos no son necesarios, pero otros son imprescindibles para conseguir las notas. En nuestro paseo nos hemos encontrado con sacacorchos, llaves, folios, guisantes, botellas, pilas, pesas y un sin fin de artículos más que es conveniente recordar donde se encuentran para localizarlos cuando los necesitemos.
Sólo es posible transportar simultáneamente los objetos, por lo que hay que seleccionarlos cuidadosamente; para conseguirlos basta con pasar por encima de ellos, sustituyendo el nuevo por alguno de los que ya teníamos.
Tommy se mueve a izquierda y derecha, salta y entra en las habitaciones, admitiendo la posibilidad de utilizar *joysticks*.
Mientras que el movimiento del personaje es bastante sencillo, la obtención de cada una de las notas supone la realización de varias actividades. De este modo para conseguir una determinada nota hay que buscar por todo el colegio dos objetos definidos previamente y



llevarlos al aula de la asignatura correspondiente. Pero a su vez la consecución de dichos objetos requiere la localización anterior de otros dos, alguno de los cuales se encuentra en una sala que para acceder a ella es necesario disponer de otro objeto... En fin, que se trara de una sucesión de actividades encadenadas que además puede romperse si te olvidas de algunos de los pasos. Es conveniente no dejarse dentro de una habitación el objeto que nos ha permitido entrar en ella, ya que en ese caso no es posible volver a dicha habitación.
En la esquina derecha aparece un billete que se va gastando a medida que **Tommy** sufre algún accidente; en el centro encontramos la puntuación, con los objetos a la izquierda de la pantalla.
Con un atractivo color y unos buenos gráficos, el programa creado por **Future Stars** cuenta con un conocida banda sonora, el tema

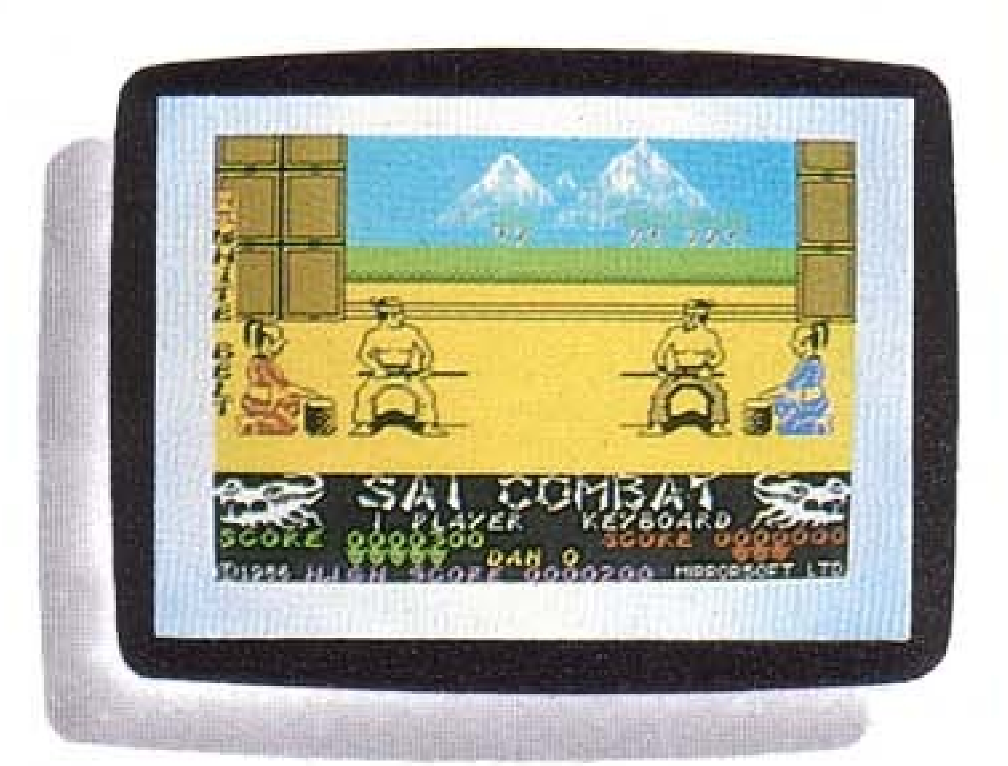


Hawai-Bombay del grupo **Mecano** que suena cuando se ha cargado el juego. El premio final para **Tommy**, según la combinación de notas, es Hawai, Egipto, Suiza o el Polo... algunos nos tuvimos que conformar con algún lugar más próximo.

****

COMBATE DE SAMURAIS

Tras las meditaciones iniciales, dos expertos luchadores van a competir en distintos escenarios orientales. Su objetivo es convertirse en el primer samurai del imperio, con todo lo que ello supone. Pero el camino hasta llegar a ser el mejor no es fácil, ya que es necesario ir ascendiendo en la escala hasta alcanzar el octavo dan, en un combate sin descanso.
Para derribar al rival es necesario golpearle en tres ocasiones, aunque con dos golpes certeros es posible dar con sus huesos en tierra. Eso sí, siempre y cuando el protagonista no doble la rodilla antes que su contrincante. Los golpes recibidos por cada contendiente vienen reflejados en las cabezas de un dragón situadas en los laterales de la pantalla que van colocándose a medida que los luchadores reciben los impactos. Se dispone de un total de seis vidas sin posibilidad de recuperarlas, perdiendo una, cada vez que el personaje termina en el suelo.
Partiendo con un cinturón blanco, se accede a la siguiente fase tras derribar en tres ocasiones al oponente y así sucesivamente, paseando por los cinturones amarillo, gris, verde, morado, rojo, azul y negro. Una vez alcanzado el cinturón negro aparece un nuevo factor que supone mayores dificultades para vencer en el combate, ya que los dos individuos situados en los laterales comienzan a lanzar indiscriminadamente peligrosos objetos, cuyo contacto provoca serios daños a los contendientes. Para aquellos que consigan derrotar al cinturón negro, continuarán su progreso hasta alcanzar el octavo dan que es el mayor grado del juego. Lógicamente cada fase sobrepasada implica enfrentarse con un luchador



DATOS GENERALES

TITULO Sai Combat

FABRICANTE Mirrorsoft

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA Artes marciales

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	5
INTERES	6
GRAFICOS	7
COLOR	5
SONIDO	7
TOTAL	30



más experimentado y por lo tanto más difícil de vencer.
El juego puede realizarse contra la máquina o entre dos competidores, con posibilidad de utilizar distintos *joysticks*.
La puntuación está en función de los golpes asestados al rival y en la parte inferior figura la más alta alcanzada.
El personaje puede dar saltos mortales hacia delante y hacia atrás, golpear con el pie, protegerse con el palo que porta y atacar frontalmente con dicho palo.
Un clásico programa de artes marciales creado por **Mirrorsoft** que cuenta con una agradable melodía oriental y típicos escenarios.

****

POKER DESCUBIERTO

Cuatro grandes jugadores de poker se dan cita para dilucidar entre ellos quien es el mejor. Todos parten con 1.000 dólares, siendo el vencedor aquel que consiga desplumar a los otros tres y embolsarse los 4.000 dólares en juego.
Se trata de un poker descubierto en el que no hay descartes, recibiéndose un total de siete cartas si se llega hasta la última apuesta.
El juego se inicia apostando un dólar cada jugador que le permiten recibir dos cartas ocultas y una tercera descubierta. A partir de ese momento se inician las apuestas, hablando aquel que tiene la carta de menor valor, por supuesto de las cuatro que están a la vista.
Todos los jugadores pueden pedir carta o apostar, pero si el anterior ya ha realizado una apuesta, hay que igualarla, subirla o pasar, en este último caso se pierde todo lo apostado previamente.
Los contrincantes de juego **Sam**, **Jim** y **Pat** son consumados expertos capaces de marcarse un «farol», es decir apostar fuerte sin tener buenas cartas, cosa que si no llegas hasta el final nunca sabrás. A medida que se desarrolla el juego hay que tratar de descubrir la psicología de los jugadores, aunque es bastante difícil.
El programa se encarga de seleccionar la mejor jugada con las cartas disponibles y por supuesto de llevar todas las puntuaciones.
Sí es conveniente saber que no hay comodines por lo que unas simples dobles parejas pueden ser consideradas una buena jugada.
Para aquellos que no les vaya bien la partida o gusten de emociones fuertes, en la habitación de al lado se puede jugar al **strip poker** con **Samantha Fox**, donde el objetivo es «conocer» un poco más a la bella contrincante.
A medida que se le van restando dólares se va desprendiendo de sus prendas, apareciendo en pantalla las distintas etapas de su *strip-tease*.

DATOS GENERALES

TITULO Samantha Fox

FABRICANTE Martech

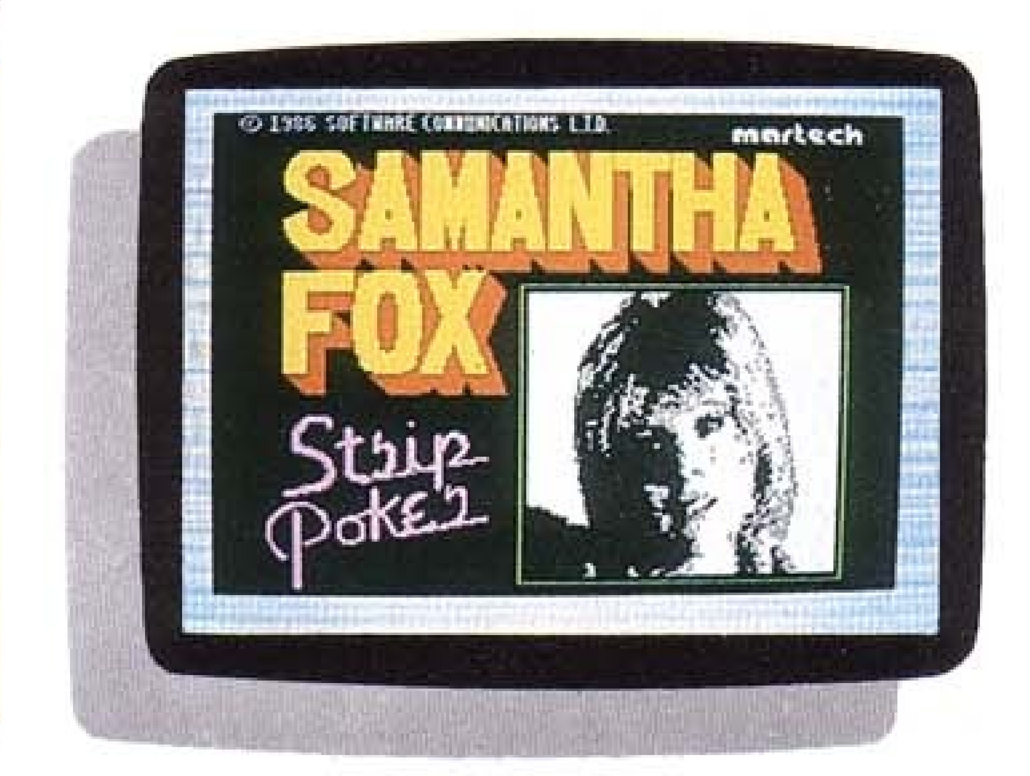
ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA Juego de naipes

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8
GRAFICOS	7
COLOR	3
SONIDO	10
TOTAL	40







Cada vez que alcanza cierto límite surge una nueva imágen que se mantendrá, siempre y cuando tus reservas no disminuyan, ya que en caso contrario se vuelve al gráfico correspondiente a los dólares disponibles.
En total son seis las imágenes de la bella **Samantha**, cada vez con más posibilidades de pillar un resfriado, hasta llegar a la última que sólo es visible cuando no la queda ni un dolar. Conseguir ganarle los mil dólares no es nada fácil, pero hemos de confesar que merece la pena intentarlo...
Por si fuera poco, este programa de **Martech** incluye, como banda sonora, una extraordinaria versión del tema principal de la película **El Golpe** que vendrá muy bien para aliviar la tensión del juego.

UN CURIOSO ASCENSOR

La popular compañía de T.V. **Network 23** dispone de un enorme edificio con un total de 210 plantas que cuentan con unas férreas medidas de seguridad.
Pero una buena técnica de emisión provoca daños físicos en los espectadores que la compañía trata de ocultar. **Edison Carter**, un intrépido reportero que ha detectado esas emisiones decide investigar en el edificio.
Una vez en el piso 200 hay que cruzar toda la planta para acceder al otro ascensor y continuar subiendo, pero esto no va a ser fácil ya que los vigilantes de planta intentarán restar energía al protagonista.
En pantalla aparece el piso en que se encuentra, la energía que le resta y el tiempo transcurrido. En los laterales se encuentran las distintas opciones: llamada al ascensor, descifrar el código, acceso al ordenador, control de cámaras, recuperación de fuerzas, apertura de puertas.
Los altos ejecutivos de la compañía tienen instaladas oficinas en los pisos 200 a 209 y para ascender a estas plantas hay que utilizar las mismas técnicas, es decir intervenir el indicador y fijar el piso deseado.
El siguiente paso es romper los códigos de seguridad para controlar las cámaras y la apertura de puertas, para ello hay que pulsar en orden inverso las cuatro notas musicales que sonarán previamente.
Tras ello es posible acceder al piso 210, lugar donde se encuentra la suite presidencial, y donde habrá de descifrar más códigos.
Otro problema que surge es el tiempo limitado ya que a las seis vuelven los trabajadores y no es posible continuar la aventura.
Se trata de un programa de **Quicksilva** que tiene como aspectos más destacados su presentación, la música inicial, los escasos, pero bien definidos objetos, y por último la originalidad de la idea.

DATOS GENERALES

TITULO Max Headroom

FABRICANTE Quicksilva

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Descifra los códigos

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	7
GRAFICOS	8
COLOR	6
SONIDO	7
TOTAL	35





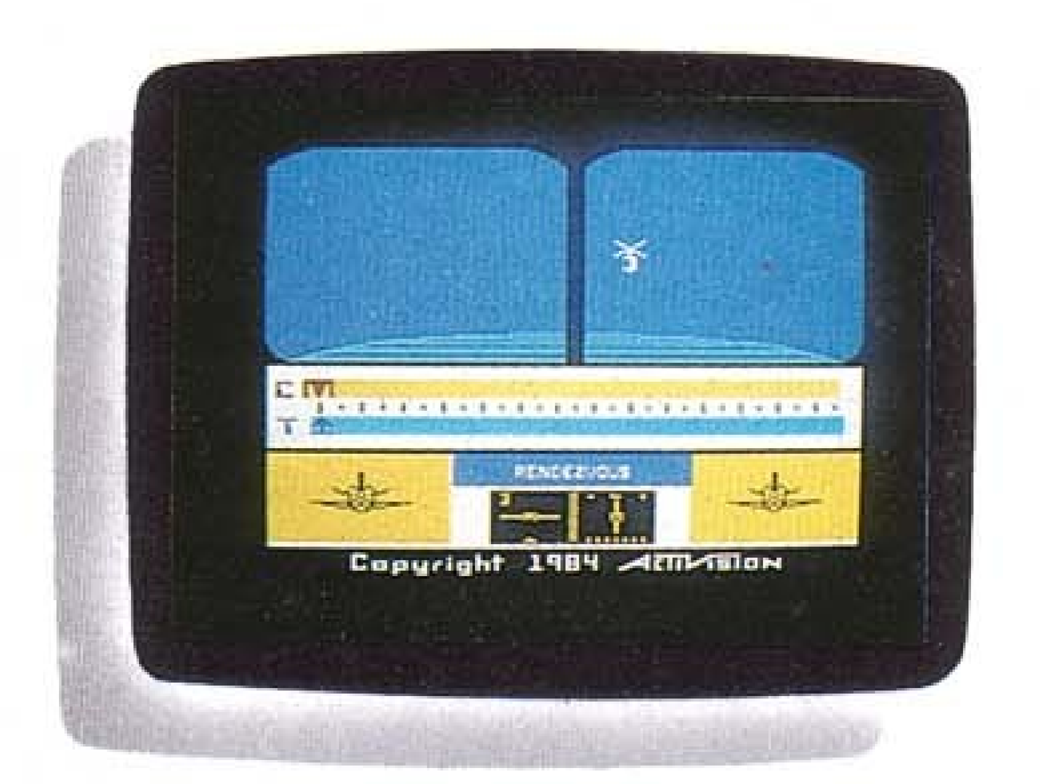


****

EL TRANSBORDADOR ESPACIAL

Una de las aplicaciones científicas más habituales del ordeandor son las simulaciones mediante las cuales es posible trasladar a un monitor un escenario real. En aviación, los pilotos pueden realizar vuelos simulados desde tierra con la ayuda de un ordenador y un buen programa.
El **Spectrum** también ofrece posibilidades y como ejemplo aquí presentamos el transbordador espacial, **Space Shuttle Discovery**, una sofisticada nave de alta precisión y difícil manejo.
La misión de la nave es contactar y atracar en un satélite orbital situado a 210 millas naúticas sobre la tierra que ha sufrido una avería y es necesario reparar.
Hay que realizar contactos y ataques como se pueda y volver a la Tierra con el menor gasto de combustible posible; según las misiones realizadas y el combustible consumido se obtendrá la calificación de comandante, piloto, especialista de misión o especialista de instrumentos.
Antes de iniciar la misión es posible seleccionar el vuelo, ofreciendo un autosimulador, simulador y misión STS 101. El autosimulador





DATOS GENERALES

TITULO Space Shuttle

FABRICANTE Activision

ORDENADOR Spectrum 48K

TEMA DEL PROGRAMA
Simulador

CALIFICACION (Sobre 10 ptos.)

ORIGINALIDAD	8
INTERES	8
GRAFICOS	5
COLOR	5
SONIDO	-
TOTAL	26

proporciona «demo» que permite conocer el mecanismo de la misión y algunos de los problemas con que nos encontramos en la realidad. El vuelo de simulador no consume combustible, y el ordenador de abordo corrige los defectos de pilotaje, pero emitiendo un mensaje cuando se produce algún error. La misión STS 101 está reservada para aquellos que conozcan suficientemente el manejo del transbordador.

Además de las cuatro direcciones habituales, existe un gran número de controles que es necesario tener en cuenta, de este modo es posible conocer la velocidad, la altitud, el combustible, el tiempo invertido, los mensajes de estado, encendido y apagado de motores, apertura y cierre de puertas de cargas y la subida y bajada del tren de aterrizaje.

En pantalla aparece la visión de espacio desde la cabina, un poco más abajo hay dos barras horizontales, una es el empuje y la otra la indicación del ordenador. A izquierda y derecha figura la imagen del transbordador y en el centro el marcador y el mapa de orientación.

Este programa realizado por **Activision** incluye un completo manual de casi 30 páginas en las que se ofrecen consejos para el lanzamiento, estabilización en órbita y aterrizaje además de unas nociones de mecánica orbital.

EL AFORTUNADO GANADOR DEL SPECTRUM PLUS RECOGE SU ORDENADOR

En el pasado número 9 de INPUT Sinclair publicabamos el nombre del ganador del sorteo de un Spectrum Plus entre quienes respondisteis a la encuesta de lectores.

Se trata de Luis Abelardo Juarez Martinez que acudió a nuestra redacción para hacerse cargo de su premio. Mas tarde decidió cambiarlo por su valor equivalente en accesorios y *software*, puesto que ya era poseedor de



ordenador propio. Enhorabuena.

¿LO HUBIERA PODIDO COMPRAR MAS BARATO?

Los clientes de Regisa esta pregunta ya no se la hacen. Pero además cuando conozcan las **nuevas ofertas** de monitores, ordenadores, impresoras, unidades de disco, periféricos, software, etc. **(evidentemente todo con garantía),** que ha preparado Regisa, se van a llevar una agradable sorpresa.

ventas al mayor

REGISA

Comercio, 11 - Tel. 319 93 08 - Barcelona

lo mismo y más..., pero al mejor precio.



sinclair AMSTRAD SPECTRAVIDEO SEIKOSHA DK-TRONIC

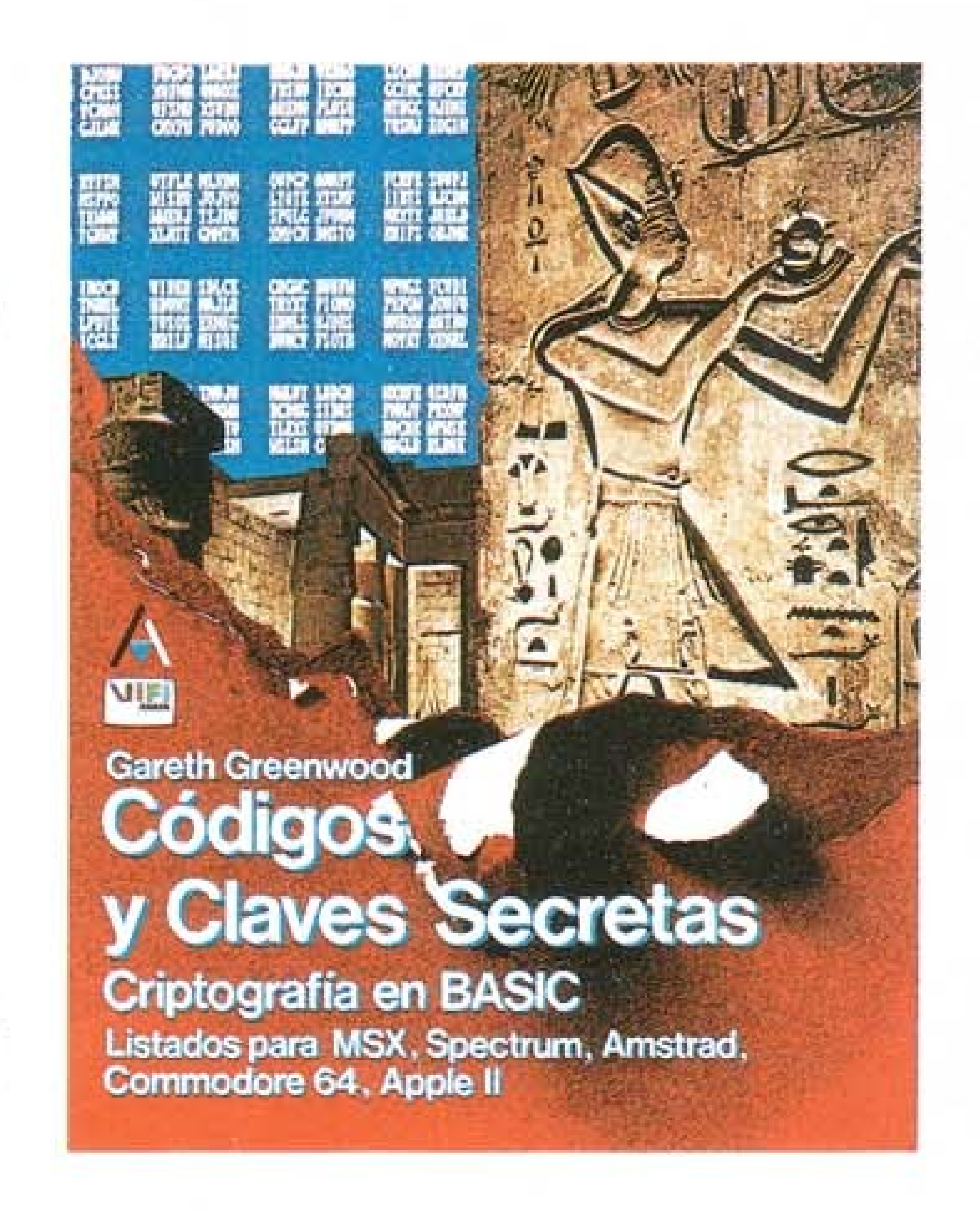
commodore HIT BIT SONY RITEMAN FONTEC

MONTERO & ASOCIADOS

Establecimientos recomendados: • BAZAR DELHI. Reina Cristina, 11. Barcelona • INTERJOYA. Reina Cristina, 9. Barcelona • BAZAR TAIWAN. Plaza Palacio, 19 (Galerías). Barcelona • LOS GUERRILLEROS. I. Canarias, 128. Valencia • BAZAR KARDIS. I. Canarias, 130. Valencia • BAZAR DELHI. M. Ruano, 5. Lleida

CONCURSO INPUT/ANAYA MULTIMEDIA

¡DESCUBRE AL ASESINO!



BASES:

Durante tres números sucesivos irán apareciendo en **INPUT** las tres partes del relato «El caso del anciano asesinado» junto con los mensajes cifrados que constituyen las respuestas parciales a la solución del crimen. Cada mes, quienes logréis descifrarlos participaréis en un sorteo consistente en 10 lotes de libros por un valor de 15.000 ptas., a elegir del fondo editorial de **Anaya Multimedia**, y una suscripción gratuita por un año a **INPUT**. Habrá tres sorteos, de tal forma que no será necesario haber descifrado el enigma del mes anterior para optar al premio.

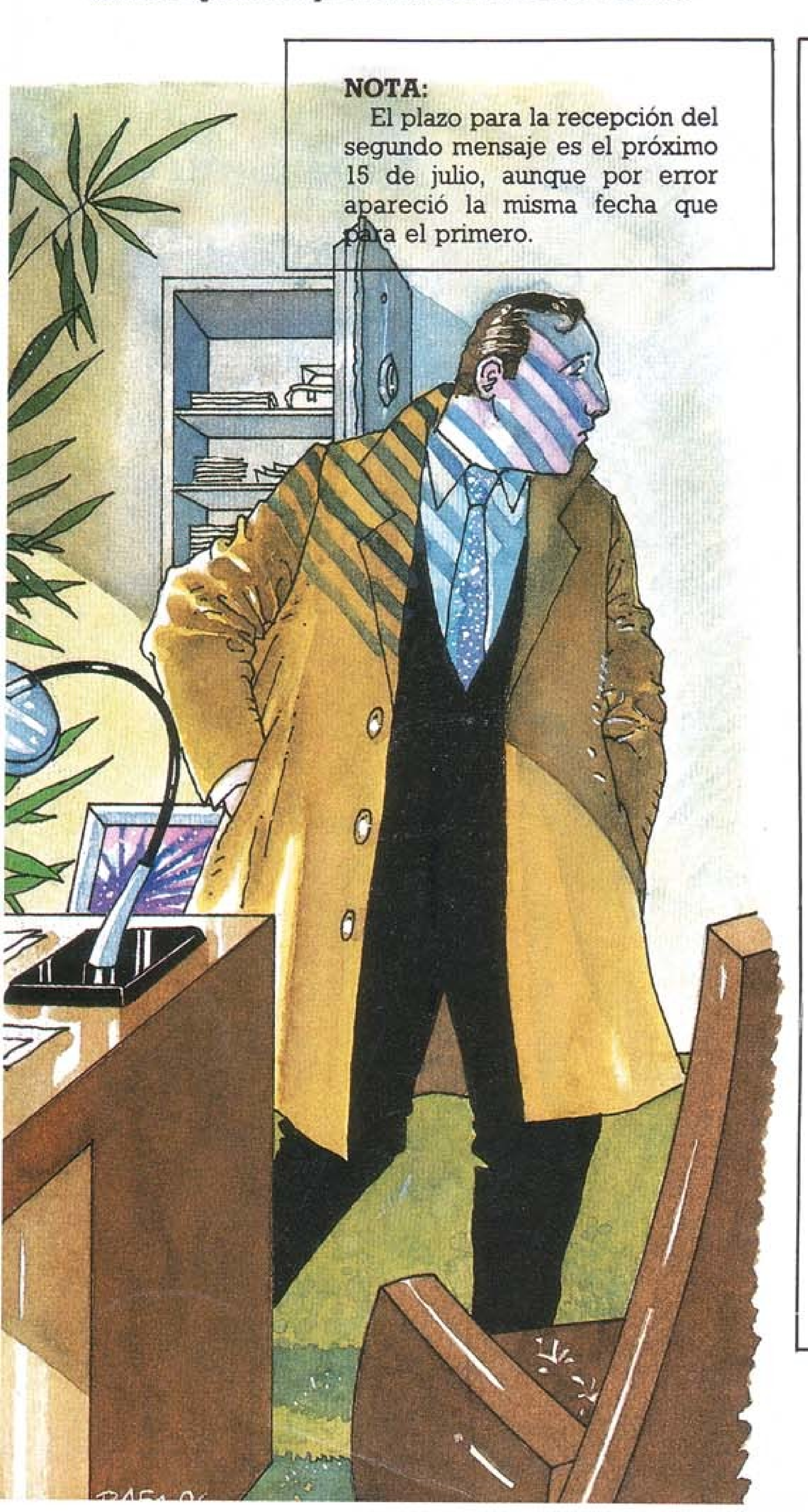
Un fabuloso premio será la guinda final de este pastel. El descubridor de las motivaciones del crimen visitará al próximo **PCW Show** a celebrar en Londres. En caso de haber más de un acertante, recurriremos a la inevitable fórmula de sorteo.

La solución del tercer mensaje cifrado deberéis enviarla, junto con vuestros datos personales, a la Redacción de **INPUT** antes del 15 de agosto.

Las decisiones del jurado serán inapelables, dándose en las páginas de **INPUT** cumplida cuenta de la marcha del concurso.

Quienes deseeis optar al premio especial debeis enviar en sobre aparte las respuestas a todos los mensajes. La fecha límite es también el 15 de agosto. El afortunado recibirá la noticia por teléfono, ya que el PCW Show *se celebra a principios de septiembre. Su nombre aparecerá publicado en INPUT de octubre.*

NOTA:
El plazo para la recepción del segundo mensaje es el próximo 15 de julio, aunque por error apareció la misma fecha que para el primero.



EL CASO DEL ANCIANO ASESINADO

—Me siento culpable —decía entre lágrima y lágrima—. Si yo hubiera estado con él, tal vez no habría muerto.

Intenté consolarle, pero la verdad es que el viejo me producía cierta repugnancia. Anoté:

```
CSQESOXDO MSPBKNY JSQ JKQ DWVOABKSELBKNV
CEKKRMODBOCCYYOZBCOXCONMKCBPXROBOYNBYVMB
ESKNVVMOYVVSKWCCKKKWSQKDC
```

Por lo demás, también vivía solo, como su hermano. Al parecer, los dos eran los últimos miembros de una familia de rancio abolengo, cuya grandeza había entrado en franco declive. De sus antiguas riquezas apenas quedaban ya unos restos: tres cuadros de firmas importantes, una pequeña casa junto al mar que casi no podían mantener y las joyas que habían robado de la caja fuerte de su hermano.

—Esto es el fin —dijo el pobre viejo entre sollozos—, esto es el fin de una importante dinastía y Dios ha permitido que viva para verlo.

Cuando llegué a la comisaría, mi ayudante había salido, pero sobre la mesa de mi despacho me había dejado una nota con el siguiente texto:

```
PFDRFBKQB ZFCOXAL QOXKPMLPFZFLK FQFKBOXO
FL BKIDOBOFLZBPBQAQCRLKBXBIXZXXIZPAXQXBS
IPXRXYBCAXIBXGPIALRLBIALJLWOFOWWQFWWWLWW
WWWWWWW
```

Junto a la nota estaba la colilla que, supuestamente, había originado el incendio y que, sin duda, era la clave del enigma. Una sonrisa se dibujó en mi rostro cuando anoté:

```
RHFTHDMSD BHEQZCN FQNMREDKC D WITGW DI G
VAES HFSXCQAI RAJMEFD LR MG HOPKOSE TK U
OHKDU HJYYIRIXPV QKWFIEWHQFTYE PCV TESIF
S HGMHHBY UOV NRZ MOINSMXRZ HF AS FYODKS
S TJRZKRZS
```

... Había resuelto un nuevo caso.

Juan José Millas

Konami SHOP®

La marca nº1 en videojuegos para ordenador abre su primera tienda de Europa en Madrid



Rogamos a todos los usuarios de Spectrum, CM 64, Amstrad y MSX que nos escriban a SERMA sugiriendo ideas para hacerlas realidad en nuestra tienda

TODOS LOS JUEGOS DEL CATALOGO DE SERMA ESTARAN TAMBIEN A LA VENTA EN: Konami® LOS TITULOS EDITADOS POR Konami® PARA SPECTRUM, CM 64, AMSTRAD Y MSX SON: FABRICADOS Y DISTRIBUIDOS EN EXCLUSIVA POR SERMA.

ORDENADORES SOBRESALIENTES A PRECIOS QUE HACEN ESCUELA

DOBLE REGALO FIN DE CURSO

Premie el esfuerzo de sus hijos por fin de curso.
Regáleles los mejores ordenadores personales a precios de auténtica oportunidad.
Investrónica, además, les hace otro gran regalo: joysticks, interfaces, cursos de Basic en vídeo, lápides ópticos...
Spectrum Plus, Spectrum 128 y QL, tres ordenadores muy estudiados, a precios que son una lección magistral.
Dé un ejemplo. Haga un doble regalo fin de curso con Investrónica.

Y además, precios muy especiales para lotes de Interface I, Microdrives e impresoras. Infórmese en su concesionario Investrónica más cercano.



SPECTRUM PLUS, SPECTRUM 128 Y QL

Regale un Spectrum Plus, que incluye un lote de 6 cintas de juegos. Su distribuidor Investrónica le regala, además:
Un joystick más un Interface II,
o un Curso de Basic en vídeo,
o un lápiz óptico.

Regale un Spectrum 128, que incluye dos cintas de juegos, un manual de utilización y una cinta de demostración.
Su distribuidor Investrónica le regala, además:
Un joystick más un Interface II,
o un Curso de Basic en vídeo.

Regale un ordenador QL desde 44.550 ptas. o, si lo prefiere, una configuración de ordenador y monitor desde 65.300* ptas.
Infórmese de nuestras grandes ofertas de QL con monitores monocromo y de color con media y alta resolución e impresora.

(PROMOCION ESPECIAL POR TIEMPO LIMITADO)

* Precio sin IVA.



investronica

Tomás Bretón, 62.
Tel. (91) 467 82 10.
Telex 23399 IYCO E.
28045 Madrid

Camp, 80.
Tels. (93) 211 26 58 - 211 27 54.
08022 Barcelona

etc